Summer 1985

The Unification and Decomposition of Processing Structures Using Lattice Theoretic Methods

David L. Livingston
Old Dominion University

THE UNIFICATION AND DECOMPOSITION OF PROCESSING STRUCTURES USING LATTICE THEORETIC METHODS

by

David L. Livingston
B.S.E. December 1976, Old Dominion University
M.S. December 1978, Old Dominion University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Electrical Engineering

OLD DOMINION UNIVERSITY
August, 1985

Approved by:

_____
Dr. Murali R. Varanasi (Director)

_____

_____

_____

ABSTRACT

THE UNIFICATION AND DECOMPOSITION OF
PROCESSING STRUCTURES USING
LATTICE THEORETIC METHODS.

David L. Livingston
Old Dominion University, 1985
Director: Dr. Murali Varanasi

The purpose of this dissertation is to demonstrate that lattice theoretic methods can be used to decompose and unify computational structures over a variety of processing systems. The unification arguments provide a better understanding of the intricacies of the development of processing system decomposition. Since abstract algebraic techniques are used, the decomposition process is systematized which makes it conducive to the use of computers as tools for decomposition. A general algorithm using the lattice theoretic method is developed to examine the structures and therefore decomposition properties of integer and polynomial rings. Two fundamental representations, the Sino-correspondence and the weighted radix representation, are derived for integer and polynomial structures and are shown to be a natural result of the decomposition process. They are used in developing systematic methods for decomposing discrete Fourier transforms and discrete linear systems. That is, fast

Fourier transforms and partial fraction expansions of linear systems are a result of the natural representation derived using the lattice theoretic method. The discrete Fourier transform is derived from a lattice theoretic base demonstrating its independence of the continuous form and of the field over which it is computed. The same properties are demonstrated for error control codes based on polynomials. Partial fraction expansions are shown to be independent of the concept of a derivative for repeated roots and the field used to implement them.

To my wife Barbara.

Acknowledgments

I would like to acknowledge Dr. M. R. Varanasi for his encouragement and persistence, the National Aeronautics and Space Administration for their early support of this work and International Business Machines Corporation for providing the needed time and support to complete this work.

TABLE OF CONTENTS

List of Tables

List of Figures

List of Symbols

Z_N ring of integers modulo N

G_N integer addition group modulo N

M_N integer multiplication monoid modulo N

$R_{Pm(x)}$ ring of polynomials modulo Pm(x)

$\ni$ such that

$\forall$ for all

$\exists$ there exists

$\in$ belongs to or set membership

$\subseteq$ is a subset or subalgebra of

$\Rightarrow$ implies

$\rightarrow$ maps to

$\neq$ not equal to

$\leq$ partially ordered to

$\prod$ product

$\sum$ sum

π, θ, τ equivalence relations

$a \equiv b(\theta)$ a is congruent to b under the congruence θ

$|\ \ |_N$ modulo N

$\begin{bmatrix} a \end{bmatrix}$ array or matrix

Chapter I

Introduction, Background and Thesis Structure

Advancing technology is providing the means for the development of very complex processing systems. The processing capabilities of such systems have extended their use in various areas of engineering, mathematics and other sciences. The cost of increased complexity, however, is an increase in the difficulty of the system design. The system design problem consists of finding an implementation which effectively meets certain design criteria. As the system complexity grows, the number of possible implementations also grows, usually at a much greater rate. Thus, determining implementations best suited to the design criteria becomes an overwhelming task. The designer must resort to the use of automated tools and decomposition techniques to accomplish the goals defined by the system design problem.

Decomposition is a very important tool in attacking the design of complex processing systems. It reduces a complex system into a collection of subsystems each of which is more simple and manageable than the original. The problem which results from such an operation is twofold. First, there may be many possible decompositions. Thus, the decompositions

which best meet the design criteria must be identified. Second, the decomposition process is often performed using intuitive or heuristic methods. These methods rely heavily on the designer's experience and capacity for innovation. It is therefore desirable to know something of the underlying structure of the system under study and to obtain this information in a systematic manner.

Abstract algebraic techniques have been used to augment the needed systematicity to a variety of decomposition problems. In particular, equivalence classes or partitions on a given algebra are often used. We conjecture that for every decomposition of a problem there is an associated set of partitions from which the decomposition may be derived. When the partitions on a given problem are related in a partial ordering, a structure theory can be developed to study the properties of the problem. Algebraic structure theories have been developed for the study of decomposition properties of combinational logic systems [1][2] and sequential machines [2][3]. The partial orderings used to develop a structure theory are of a particular type called a lattice. It is lattice theory and an offshoot, universal algebra theory, which are fundamental to the study of decomposition.

It is the purpose of this dissertation to show that lattice theory and universal algebra theory can be used to develop general structure theories and thus decomposition

properties for a variety of topics in the study of processing systems. According to Garrett Birkhoff, considered the father of lattice theory, "In general, lattice theory has helped to simplify, unify and generalize many aspects of mathematics..." [4]. Since mathematics is the foundation upon which the design of processing systems is based, not to mention the sciences of engineering and physics, the use of lattice theory in the design process can have the same simplifying, unifying, and generalizing effects. It will be demonstrated that lattice theory can be used to derive many results in the theory of processing systems in a simple and systematic manner. It will also be demonstrated that there are isomorphic structures across the variety of subjects. Thus, the application of lattice theory serves to unify the structure properties of different systems.

Background

Lattice theory is a relatively new branch of mathematics. Therefore its applications to engineering are still being developed. The science of computer and digital system design has used many tools obtained from abstract algebra including lattice theory.

R. L. Ashenhurst developed a theory for the decomposition of switching circuits which is included in a book by H. A. Curtis [1]. The principle behind Ashenhurst's theory is the construction of partition matrices on the

variables for a given switching function. The matrices with column multiplicity $\leq$ 2, i. e., no more than two distinct columns, yield decompositions of the switching function over its variables. The column multiplicity criterion conforms to the substitution property which will be discussed in the succeeding material. Two-dimensional partition matrices, i. e., partitions containing two blocks, yield disjunctive decompositions. Three block partitions are used to obtain nondisjunctive decompositions. Ashenhurst showed that the various decompositions of a given switching function form a lattice from which structure information is obtained.

J. Hartmanis and R. E. Stearns published a number of papers on the structure of sequential machines which were compiled into a book entitled Algebraic Structure Theory of Sequential Machines [3]. The techniques developed by Hartmanis and others consist of finding lattices of partitions on the state, input and output sets of a sequential machine. The structure or information flow can be determined from these lattices and various decompositions possessing desirable properties can be obtained. These techniques are a very direct use of lattice theory and are an integral part of the method of decomposition and structure theory proposed in this dissertation. Thus, they will be discussed in more detail in a subsequent section.

An area in which decomposition plays an important role is computational complexity. In computational complexity theory, various algorithms for implementing the solution to a given computational problem are studied with the goal of finding the algorithm requiring the least number of calculations to solve the problem. One method for obtaining the algorithms to be studied is decomposition of the problem into elementary operations. The operations can then be weighted and optimization algorithms can be applied. The Fast Fourier Transform [5] and Winograd Fourier Transform [6] algorithms are good examples of studies in complexity theory. Another example is the complexities of integer and polynomial arithmetics as studied by S. Winograd [7][8].

Unification of the theories and methods used in differing subjects is an important concept. It allows the transfer of properties from one subject to the other and also provides a singular method by which results can be obtained regardless of the subject. This contributes to a better understanding of the subjects and simplifies the design processes which are contained within them. R. Blahut published a paper which demonstrates that digital signal processing and error control coding have a common basis in the discrete Fourier transform and argues that both subjects are theoretically independent of the cardinality of the fields over which they are defined [9]. Through the use of lattice theory, we will demonstrate that similar results can

be obtained from a basis which is more fundamental than the discrete Fourier transform.

Structure

This dissertation shows that lattice theoretic techniques can be used to develop a unified structure theory for various types of processing systems. This is accomplished in a progressive manner; i. e., concepts which are introduced in each chapter are used in succeeding chapters.

Chapter II provides a review of the mathematical background needed to apply lattice theoretic methods to processing systems. The theories of partitions, lattices, universal algebras, and their use in the decomposition process are examined.

Chapter III applies the theoretical basis developed in Chapter II to formulate a structure theory for finite integer operations. Finite integer rings are shown to be examples of universal algebras. From the lattices associated with the algebras, a general decomposition method is derived.

Chapter IV is analogous to Chapter III with the exception that finite polynomial structures are examined. Similarities between the structures of integers and polynomials are discussed. Results relating to the discrete Fourier transform and error control codes are also developed.

Chapter V applies the structure theories developed in Chapters III and IV to the discrete Fourier transform and linear systems. It is demonstrated that lattice theoretic techniques can be used to derive fast Fourier transform algorithms and partial fraction expansions of linear systems.

Chapter VI summarizes the dissertation and concludes with recommendations for possible research.

Chapter II

Lattice Theory, Partitions and Universal Algebras

The following is a brief review of the theories of lattices, partitions, and universal algebras. More detail may be found in references: [10][11][12][13][14][15]. Knowledge of elementary set theory and algebraic structures is assumed.

Lattices

A set with a relation $\{P, \leq\}$ is called a partially ordered set or poset if $\forall$ a, b, c $\in$ P, $\leq$ has the following three properties:

1) reflexivity, $a \leq a$;

2) antisymmetry, $a \leq b$ and $b \leq a \Rightarrow a = b$ and

3) transitivity, $a \leq b$ and $b \leq c \Rightarrow a \leq c$.

Given a poset $\{P, \leq\}$ and a subset $Q \subseteq P$, an element $p \in P$ is called the greatest lower bound (glb) if and only if:

1) $p \leq q$, $\forall\, q \in Q$ and

2) $r \leq q$, $\Rightarrow r \leq p$, $\forall\, q \in Q$ and $r \in P$.

By the duality principle an element $p \in P$ is called the least upper bound (lub) if and only if:

1) $q \leq p$, $\forall\, q \in Q$ and

2) $q \leq r$, $\Rightarrow p \leq r$, $\forall\, q \in Q$ and $r \in P$.

A poset $\{L, \leq\}$ with a glb and lub for every pair of elements belonging to L is called a lattice. A lattice can also be defined by the triple $\{L, +, \cdot\}$ where L is a set and $\forall$ a, b, c $\in$ L, + and $\cdot$ satisfy the following four properties:

1) idempotentcy, $a + a = a$ and $a \cdot a = a$;

2) commutativity, $a + b = b + a$ and $a \cdot b = b \cdot a$;

3) associativity, $a + (b + c) = (a + b) + c$ and

$$a \cdot (b \cdot c) = (a \cdot b) \cdot c;$$ and

4) absorption, $a + (a \cdot b) = a$ and

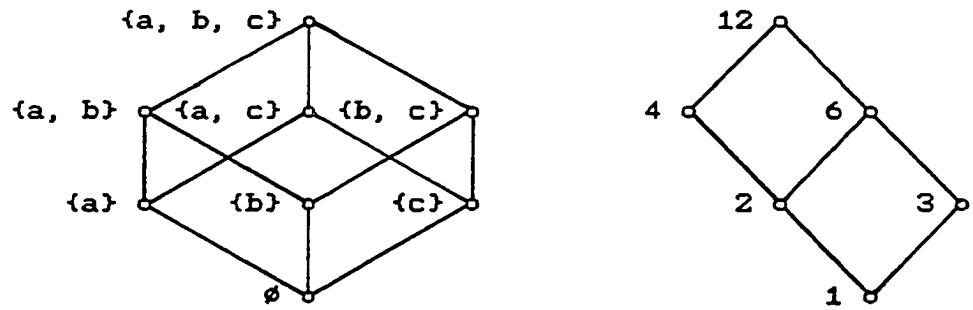
$$a \cdot (a + b) = a.$$

It can be shown that the descriptions of a lattice, $\{L, \leq\}$ and $\{L, +, \cdot\}$, are equivalent and that the relations which produce the lub and glb are equal to + and $\cdot$, respectively. In a finite lattice L there exist two elements typically denoted by I and O such that $\forall$ a $\in$ L:

1) $I + a = I$ and $I \cdot a = a$ and

2) $O + a = a$ and $O \cdot a = O$.

Lattices can be represented in a graphical form called a Hasse diagram. Nodes in the diagram represent elements of the poset while edges represent the relation between the elements. Two examples of lattices and their associated Hasse diagrams are the power set P(a, b, c), Figure 2.1a, and the lattice of all integer divisors of 12, Figure 2.1b. A subset M of a lattice L is called a sublattice if $\forall$ a, b $\in$ M, a + b and a $\cdot$ b $\in$ M.



(a) (b)

Figure 2.1. Examples of Lattices.
(a) Power set P(a, b, c). (b) Divisors of 12.

Lattices are classified into many categories according to the properties they exhibit. Two important categories are distributive and modular lattices. A lattice L is distributive if and only if the following property is satisfied $\forall$ a, b, c $\in$ L:

1) $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ and

$a + (b \cdot c) = (a + b) \cdot (a + c)$.

A lattice L is modular if and only if $\forall$ a, b, c $\in$ L:

1) $a \leq c \Rightarrow a + (b \cdot c) = (a + b) \cdot c$.

All distributive lattices are modular; however, the converse is not true. Examples of lattices which are nondistributive and nonmodular are shown in Figure 2.2a and Figure 2.2b. It can be shown that a lattice L is not distributive if and only if either or both of the lattices in Figures 2.2a and 2.2b are sublattices of L. Similarly, a lattice is not modular if and only if it contains the lattice of Figure 2.2b. Categorizing lattices as to their properties can be

(a) (b)

Figure 2.2. Nonmodular and Nondistributive Lattices. (a) Nondistributive lattice. (b) Nonmodular Lattice.

useful in generalizing the structure of various classes of algebras. For example, the lattice of all normal subgroups of a group is always a modular lattice.

Partitions

Given a set S, a relation on S denoted Θ is called an equivalence relation if Θ has the following properties $\forall$ a, b, c $\in$ S:

1) reflexivity, a Θ a;

2) symmetry, a Θ b $\Rightarrow$ b Θ a and

3) transitivity, a Θ b and b Θ c $\Rightarrow$ a Θ c.

Note that a Θ b is typically written as a $\equiv$ b (mod Θ) or just a $\equiv$ b (Θ). Given {S, Θ}, $\forall$ a $\in$ S, the set B_a = {b | a $\equiv$ b (Θ)} is called an equivalence class defined by a. A set of blocks P on a set S is called a partition if:

1) $B_i \cap B_j = \emptyset$, $\forall$ B_i, $B_j \in P$ and

2) $\cup P = S$.

A partition is typically represented by a set of blocks denoted by B_i which are separated by semicolons and are further emphasized by overbars. Elements contained in the blocks are separated by commas. Partitions are usually denoted by lowercase Greek letters π, τ, etc. For example, given the set

$S = \{1, 2, 3, 4, 5, 6, 7, 8\}$,

$\tau_1 = \{\overline{1, 2}; \overline{3, 4}; \overline{5, 6, 7}; \overline{8}\}$ and

$\tau_2 = \{\overline{1}; \overline{2, 6, 7}; \overline{3, 8}; \overline{4}; \overline{5}\}$

are two partitions of S. For any finite set S there are always two trivial partitions: π_I, where all elements belonging to the set S are contained in one block; and π_0, where each element belonging to the set S is contained in its own block.

Partitions can be partially ordered. Let P be a set of partitions; two partitions $\tau_i, \tau_j \subseteq P$ are related as $\tau_i \leq \tau_j$ if and only if every block of τ_i is a subset of some block of τ_j. For example, given the partitions:

$\tau_3 = \{\overline{a, b}; \overline{c, e}; \overline{d}; \overline{f}\}$ and

$\tau_4 = \{\overline{a, b}; \overline{c, d, e}; \overline{f}\}$

then $\tau_3 \leq \tau_4$.

In accordance with the partial ordering, partition "addition" and "multiplication" can be defined. Given two partitions τ_i and τ_j, their sum τ_k may be obtained by taking the union of blocks belonging to τ_i and τ_j if their intersection is not empty. That is, if

$x \equiv y(\tau_i)$ and $y \equiv z(\tau_j)$ then

$\tau_i + \tau_j \Rightarrow x \equiv y \equiv z(\tau_k)$.

Multiplication of partitions is a simpler operation. The product of two partitions, $\tau_i \cdot \tau_j = \tau_k$, is obtained by the intersection of the blocks of τ_i and τ_j. That is, if

$x \equiv y(\tau_i)$ and $x \equiv z(\tau_j)$ then

$\tau_i \cdot \tau_j \Rightarrow x \equiv x(\tau_k)$.

For example, given partitions

$\tau_5 = \{\overline{1};\ \overline{2,\ 3,\ 4};\ \overline{5,\ 6};\ \overline{7,\ 8}\}$ and

$\tau_6 = \{\overline{1,\ 8};\ \overline{2,\ 5,\ 6};\ \overline{3,\ 4};\ \overline{7}\}$, then

$\tau_5 + \tau_6 = \{\overline{1,\ 7,\ 8};\ \overline{2,\ 3,\ 4,\ 5,\ 6}\}$ and

$\tau_5 \cdot \tau_6 = \{\overline{1};\ \overline{2};\ \overline{3,\ 4};\ \overline{5,\ 6};\ \overline{7};\ \overline{8}\}$.

Note that the trivial partitions π_I and π_ϕ are multiplicative and additive identities, respectively.

Given the partial ordering defined above and restricting the discussion to finite sets, the set of all partitions of a finite set forms a lattice. In such a lattice, the results of partition addition and multiplication are the lub and glb, respectively. The partition lattice is an important one, for it describes the "structural universe" of a given set and is therefore commonly referred to as the structure lattice. Sublattices of the structure lattice composed of partitions with certain properties are the basis upon which a systematic decomposition method is derived. The partition lattice

grows very large as the number of elements in the set operated on by the equivalence relations increases. The number of partitions on a finite set with n elements is called a Bell number B_n.

$$B_n = \sum_{k=1}^{n} S_{n,k}; \qquad (2.1)$$

where n is the number of elements in the set, k is the number of blocks in a partition and $S_{n,k}$ are Stirling numbers of the second kind. Bell numbers can easily be obtained by the recursion:

$$B_0 = 1;\ B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k. \qquad (2.2)$$

As an indication of how fast the number of partitions grows versus the number of set elements, the Bell number $B_4 = 15$ whereas $B_8 = 4140$. In this particular case doubling the number of set elements produced a 276-fold increase in the number of partitions on the set. It is important to remember, however, that only a subset of these partitions produce decompositions with desirable characteristics.

Universal Algebras

A universal algebra, or simply algebra, is defined to be a two-tuple: {S, F}. S is a nonempty set which will be referred to as the operand set and F is a set of mappings which will be referred to as the mapping or operator set such that $\forall\ f\alpha \in F$, $f\alpha: S^{n(\alpha)} \rightarrow S$, where $n(\alpha)$ is a nonnegative integer and $S^{n(\alpha)}$ is the Cartesian product of S,

$n(\alpha)$ times. Structures such as groups and rings are examples of universal algebras. Note that a field is not an algebra as defined above since the multiplicative inverse mapping is not defined for the additive identity. An algebra $\{T, F\}$ is called a subalgebra of $\{S, F\}$ if $T \subseteq S$ and $\forall f\alpha \in F$, $f\alpha$: $T^{n(\alpha)} \rightarrow T$. That is, the Cartesian product of T $n(\alpha)$ times is closed in T under $f\alpha$. The subalgebras of an algebra form a lattice under the partial ordering of set inclusion. The glb of two subalgebras is simply the set intersection of the operand sets belonging to the subalgebras. The lub of two subalgebras is the set of images of the mappings belonging to the operator set on the elements belonging to the union of the operand sets. For example, Figure 2.3 is the lattice of subgroups of the addition group modulo six.

Given an algebra $\{S, F\}$, an equivalence relation Θ on $\{S, F\}$ is called a congruence relation if $\forall f\alpha \in F$,

$$a_i \equiv b_i \ (\Theta) \Rightarrow$$

$$f\alpha(a_1, \ldots, a_{n(\alpha)}) \equiv f\alpha(b_1, \ldots, b_{n(\alpha)}) \ (\Theta),$$

for $i = 1, \ldots, n(\alpha)$. The property which distinguishes a congruence from an equivalence relation is called the substitution property. Since partitions are representations of equivalence relations, some partitions are congruences and hence have the substitution property. These partitions are called substitution property partitions or

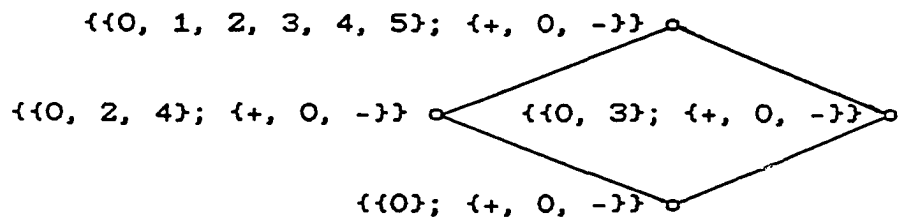


Figure 2.3.
Lattice of Subgroups of the Addition Group G_6.

S. P. partitions and are very important in determining the underlying structure of a given algebra. The set of congruences on an algebra forms a lattice which is a sublattice of the lattice of all equivalence relations on the algebra. That is, the lattice of S. P. partitions is a sublattice of the structure lattice of a given algebra.

Decomposition of Algebras

Given two algebras with the same mapping set, {R, F} and {S, F}, their direct product can be obtained by taking elements from the Cartesian product of their operand sets

$$R \times S \ni \forall f\alpha \in F,$$

$$f\alpha: (R \times S)^{n(\alpha)} \rightarrow (R \times S),$$

where $n(\alpha)$ is a nonnegative integer. That is, $\forall r \in R$ and $s \in S$,

$$f\alpha([r_1, s_1], \ldots, [r_{n(\alpha)}, s_{n(\alpha)}]) =$$

$$[f\alpha(r_1, \ldots, r_{n(\alpha)}), f\alpha(s_1, \ldots, s_{n(\alpha)})].$$

Thus, algebras may be composed from the direct product of others. Given a direct product $\Pi Ai = \Pi\{Si, F\}$, a subalgebra of ΠAi, $B = \{R, F\}$, is called a subdirect product if $\forall s_i \in Si$, $\exists r \in R \ni s_i$ is the ith component of r. A direct

product is trivially a subdirect product of itself. A homomorphism which maps an algebra into its subdirect product of similar algebras is called a representation of the algebra as the subdirect product of its component algebras.

The following results are summarized from an important theorem and its corollary from Birkhoff. The actual theorem statement and proof are given in [10], page 140.

Theorem 2.1. Given a representation of an algebra A as a subdirect product C of similar algebras Ai, then C is isomorphic to the factor algebra $A/\Pi\Theta i$ over the product of congruences derived from the homomorphism mapping A onto $Ai = A/\Theta i$. The converse also holds. •

Corollary 2.1. The isomorphic representations of an algebra as a subdirect product correspond injectively to the sets of congruence relations on the algebra such that the product of the congruence relations is the equality relation. •

From the above statements, a method for the composition of algebras is evident. For example, given the groups

$G_2 = \{\{0, 1\}; \{+, 0, -\}\}$ and

$G_3 = \{\{0, 1, 2\}; \{+, 0, -\}\}$,

the direct product is

$G_2 \times G_3 =$

$\{\{(0,0), (0,1), (0,2), (1,0), (1,1), (1,2)\};\{+, 0, -\}\}$.

It is obvious that $G_2 \times G_3$ is isomorphic to G_6.

The inverse operation, decomposition, follows from another theorem, again from Birkhoff [10], page 164:

Theorem 2.2. The direct decompositions of an algebra are obtained from the sets of congruences Θi on the algebra such that their product $\Pi\Theta i$, for $i = 1, \ldots, n$, is the equality relation and for $i = 2, \ldots, n$,

1) $\Theta 1 \cdot \ldots \cdot \Theta i-1$ is permutable with Θi and

2) $(\Theta 1 \cdot \ldots \cdot \Theta i-1) + \Theta i = I$, where I is the trivial congruence of tautology. ∎

In terms of partition notation, the above theorem implies that if the product of S. P. partitions on an algebra equals the trivial S. P. partition π_ϕ, then the algebra can be directly decomposed. Statement 2 in Theorem 2.2 insures that the representations obtained are not further directly decomposable. Examples of how Theorem 2.2 is used to obtain a decomposition are demonstrated in the succeeding material.

Theorem 2.2 suggests a general method for the direct or parallel decomposition of an algebra.

The Parallel Decomposition Method:

1) Find all S. P. partitions on the algebra under decomposition.

2) Form the Hasse diagram of the lattice of S. P. partitions. The diagram assists in determining which partitions should be chosen to obtain a desirable decomposition.

3) Choose a set of S. P. partitions which meet the

criteria of Theorem 2.2.

4) Based on the partitions selected, choose a homomorphism which maps the algebra into a representation.

Step 4 is an intuitive process. Given the partitions for which the product is the equality relation, each block of each partition is assigned, via the homomorphism, to an element which belongs to an algebraic system. This algebraic system is typically isomorphic to a subalgebra of the algebra under decomposition which directly corresponds to the partition. This particular assignment is called the natural representation. Other assignments can be chosen according to a given design criteria. The process of assigning blocks or choosing homomorphisms will become apparent in succeeding examples. Figure 2.4 is a block diagram of a direct or parallel decomposition.

So far, the discussion has only been concerned with direct or parallel decompositions. Given equivalence relations which are not congruences, Theorem 2.2 can still be used to obtain decompositions. This is possible because the lattice of S. P. partitions is a sublattice of the lattice of all partitions. The 0 element of the lattice of all partitions is π_0, the equality relation. Therefore if any set of partitions can be found such that their product is equal to π_0, then decompositions can be derived.

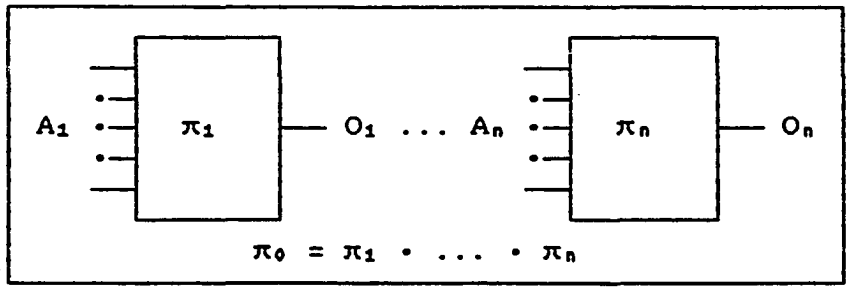


Figure 2.4.
Parallel Decomposition.

Decompositions of this type, however, may have unpredictable or undesirable properties.

Notation: S. P. partitions will be represented by the Greek letter π and partitions without the substitution property will be represented by the Greek letter τ.

Serial decompositions are a class of decompositions in which partitions without the substitution property are used. Given a set of partitions containing at least one S. P. partition such that it satisfies Theorem 2.2, then a serial decomposition can be obtained if the following property is satisfied.

The Serialization Property:

Given the product,

$$\pi_1 \cdot \tau_1 \cdot \ldots \cdot \tau_n = \pi_0,$$

then it can be permuted such that

$$\pi_1 \cdot \tau_1 = \pi_2,$$

$$\pi_2 \cdot \tau_2 = \pi_3, \ldots,$$

$$\pi_n \cdot \tau_n = \pi_0.$$

That is, starting with a congruence, each step of the

product of a congruence and an incongruent equivalence relation produces a congruence. This is represented in a block diagram in Figure 2.5.

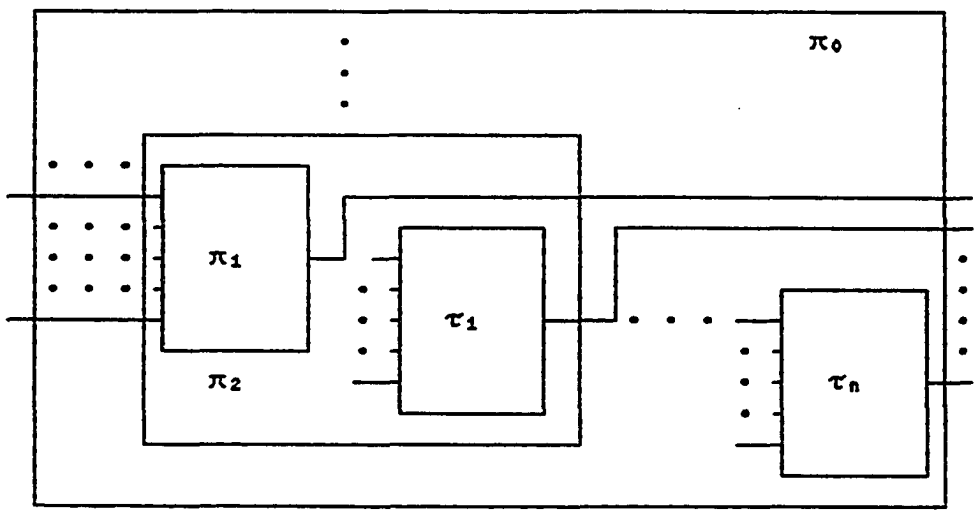


Figure 2.5.
Serial Decomposition.

Possible serial decompositions are easily identified from the lattice of S. P. partitions. A method for obtaining the serial decomposition of an algebra can be obtained from the preceding property.

The Serial Decomposition Method:

1) Find all S. P. partitions on the algebra under decomposition.

2) Form the Hasse diagram of the lattice of S. P. partitions.

3) Identify chains of S. P. partitions.

4) Choose partitions that do not have the substitution

property such that the serialization property is satisfied.

5) Select a homomorphism which maps the algebra to a representation.

For example, see Figure 2.6.

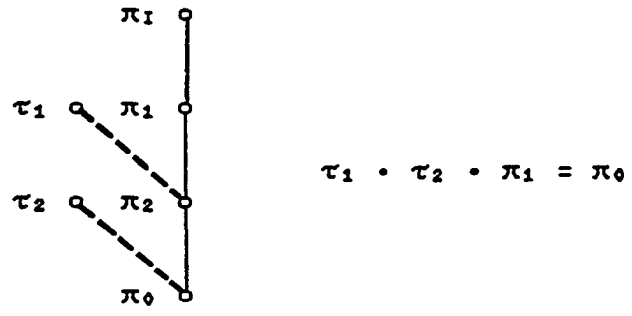


Figure 2.6.
Chain Decomposition.

The details of how serial and parallel decompositions affect algebraic structures wil . be discussed further in context.

Chapter III

Integer Decomposition

Integer arithmetic is fundamental to the operation of arithmetic processing systems. Integer addition has long been considered a primitive operation in the same category as shifting, negation and logical operations. With advances in array logic, integer multiplication has also been added to this list. Other operations such as floating-point arithmetic and addressing calculations rely on integer arithmetic as the basis for their construction. Given the importance of integer arithmetic to processing, it is desirable to gain insight into its underlying structure. Information derived from this structure can be used in the determination of efficient implementations and may reveal useful information about more complex systems.

Modern processing systems are finite machines composed of a limited amount of discrete logic. Only a finite number of digits can be operated on at a given time. A finite number of digits and thus a finite integer arithmetic imply that modular arithmetic is being performed. For a given word-length and modulus, finite integer addition and multiplication form the commutative ring of integers modulo N, where N - 1 is the largest integer representable by a

given number of digits. Many computing systems perform calculations in radix two. This implies $N = 2^n$, where n is the number of binary digits.

The ring of integers mod N, represented $Z_N = \{S_N; \{+, 0, -, \cdot, 1\}\}$, consists of an Abelian group over addition and a monoid over multiplication. Groups, and rings are examples of universal algebras with properties which are useful to the study of the decomposition of systems.

The Structure of the Integer Addition Group Modulo N

A group falls within the classification of a universal algebra $G = \{S, F\}$. The set of mappings F consists of $\{f\alpha, f\beta, f\delta\}$ where:

$f\alpha$: $S^2 \rightarrow S \ni f\alpha(a, b) = ab, \forall a, b \in S$;

$f\beta$: $S \rightarrow S \ni f\beta(a) = a, \forall a \in S$; and

$f\delta$: $S \rightarrow S \ni f\delta(a) = -a, \forall a \in S$.

As previously stated, the subalgebras of a universal algebra form a lattice. Hence, the subgroups of the integer addition group modulo N, $G_N = \{S_N; \{+, 0, -\}\}$, form a lattice. In order to derive the structure and thus the decomposition properties of G_N, it is necessary to show the relation between subgroups of G_N and congruences on G_N. We therefore present and prove the following lemma and theorem to demonstrate this relation.

Lemma 3.1. The cosets of a subgroup H of an Abelian group G have the substitution property under the group and inverse operations.

Proof.

Given

$g_i, g_j, g_k, g_l, g_\alpha, g_\beta \in G$ and

$h_i, h_j, h_k, h_l \in H$, where $H \subseteq G$,

then $g_i = g_\alpha h_i$ and $g_j = g_\alpha h_j$ imply that g_i and g_j are in the same coset. That is, they have the same coset leader g_α. Similarly, g_k and g_l are in the same coset if $g_k = g_\beta h_k$ and $g_l = g_\beta h_l$.

Let

$$g_i g_k = g_\alpha h_i g_\beta h_k \text{ and } g_j g_l = g_\alpha h_j g_\beta h_l.$$

Since G is Abelian,

$$g_\alpha h_i g_\beta h_k = g_\alpha g_\beta h_i h_k \text{ and } g_\alpha h_j g_\beta h_l = g_\alpha g_\beta h_j h_l.$$

Hence,

$$g_i g_k = g_\alpha g_\beta h_i h_k \text{ and } g_j g_l = g_\alpha g_\beta h_j h_l.$$

Since $h_i h_k$ and $h_j h_l$ are both closed by the definition of a subgroup and $g_i g_k$ and $g_j g_l$ have the same coset leader $g_\alpha g_\beta$, $g_i g_k$ and $g_j g_l$ belong to the same coset. Therefore, the cosets of H have the substitution property for the group operation. For the inverse operation, let $g_i = g_\alpha h_i$ and $g_j = g_\alpha h_j$.

Then

$$g_i^{-1} = (g_\alpha h_i)^{-1} \text{ and } g_j^{-1} = (g_\alpha h_j)^{-1}.$$

But

$$g_i^{-1} = g_\alpha^{-1} h_i^{-1} \text{ and } g_j^{-1} = g_\alpha^{-1} h_j^{-1}.$$

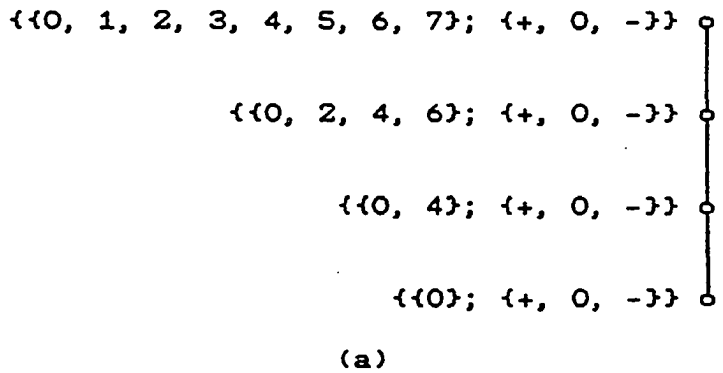
Since g_i^{-1} and g_i^{-1} have the same coset leader, g_α^{-1}, and h_i^{-1} and h_j^{-1} are closed in H, g_i^{-1} and g_j^{-1} are in the same coset and the inverse operation possesses the substitution property. ∎

Theorem 3.1. The lattice of subgroups of $G_\parallel$ is isomorphic to the lattice of S. P. partitions on $G_\parallel$.

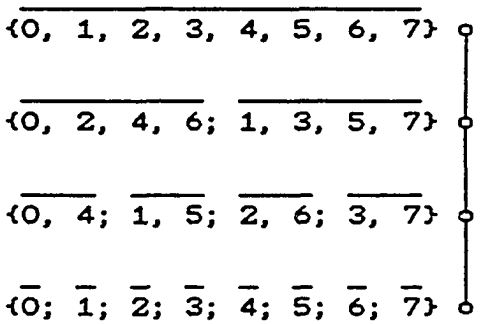
Proof. Given a subgroup of $G_\parallel$, partitions can be created by forming blocks from the cosets of the subgroup. By Lemma 3.1 the cosets and hence, the partitions, have the substitution property. The partial ordering on the lattice of subgroups is set inclusion. If a subgroup $H_\parallel$ is contained in another subgroup $J_\parallel$, then the cosets of $H_\parallel$ are also contained in those of $J_\parallel$. This is also the definition of partition inclusion which is the partial ordering in the lattice of partitions. Since the set of subgroups of $G_\parallel$ are isomorphic to the set of S. P. partitions on $G_\parallel$ and the partial orderings are isomorphs, the lattices are isomorphic. ∎

For example, Figures 3.1a and 3.1b are the isomorphic lattices of the subgroups of G_8 and the S. P. partitions on G_8, respectively.

Theorem 3.1 establishes the connection between the structure of $G_\parallel$ with its subgroups. Using the parallel and serial decomposition methods developed in Chapter II, a



(a)



(b)

Figure 3.1. Isomorphism of Subgroup and Structure Lattices.
(a) Lattice of S. P. partitions on G_8.
(b) Lattice of subgroups of G_8.

general method for the decomposition of G_8 can be obtained.

A General Method for the Decomposition of G_8.

1) Find all S. P. partitions or alternately find all subgroups and then generate the S. P. partitions from their cosets.

2) Generate the structure lattice.

3) Using the parallel and serial decomposition properties, determine which partitions should be used to derive a decomposition which meets some predetermined criteria.

4) Choose a set of representation homomorphisms.

5) Represent the decomposed group using the representation homomorphisms.

We demonstrate the above method in Example 3.1.

Example 3.1. The Parallel Decomposition of G_6.

1) We first generate the subgroups of G_6 as represented in Tables 3.1.1a - 3.1.1d. Consequently, the S. P. partitions on G_6 are

$\{\{0, 1, 2, 3, 4, 5\}; \{+, 0, -\}\} \Rightarrow \{\overline{0, 1, 2, 3, 4, 5}\} = \pi_1$;

$\{\{0, 2, 4\}; \{+, 0, -\}\} \Rightarrow \{\overline{0, 2, 4}; \overline{1, 3, 5}\} = \pi_1$;

$\{\{0, 3\}; \{+, 0, -\}\} \Rightarrow \{\overline{0, 3}; \overline{1, 4}; \overline{2, 5}\} = \pi_2$; and

$\{\{0\}; \{+, 0, -\}\} \Rightarrow \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}\} = \pi_0$.

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

$G_6 = \{\{0, 1, 2, 3, 4, 5\}; \{+, 0, -\}\}$.
(a)

+	0	2	4
0	0	2	4
2	2	4	0
4	4	0	2

$H_3 = \{\{0, 2, 4\}; \{+, 0, -\}\}$.
(b)

+	0	3
0	0	3
3	3	0

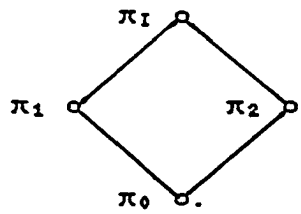
$H_2 = \{\{0, 3\}; \{+, 0, -\}\}$.
(c)

+	0
0	0

$H_1 = \{\{0\}; \{+, 0, -\}\}$.
(d)

Table 3.1.1. The Subgroups of G_6.

2) We next generate the structure lattice:



3) Select a set of partitions for which the product is π_0.

$\pi_1 \cdot \pi_2 = \pi_0$.

4) Select a set of representation homomorphisms:

Φ_1: $\{\overline{0, 2, 4}; \overline{1, 3, 5}\} \rightarrow \{\{0, 1\}; \{+, 0, -\}\} = G_2$.

Φ_2: $\{\overline{0, 3}; \overline{1, 4}; \overline{2, 5}\} \rightarrow \{\{0, 1, 2\}; \{+, 0, -\}\} = G_3$.

The homomorphisms are obtained by assigning an element to each block of each partition. Since the partitions have the substitution property, the algebras which result are groups.

5) The resulting decomposition can be expressed in tabular or block diagram form as illustrated in Tables 3.1.2a - 3.1.2b and Figure 3.2, respectively.

Note that the representation homomorphisms used in Example 3.1 are simply the residues, modulo 2 and modulo 3. By Theorem 2.1, the Cartesian product of G_2 and G_3 is isomorphic to G_6. The decomposition derived from the lattice method is essentially the Sino-Correspondence [16]. Therefore the inverse operation is the Chinese Remainder Theorem. (See Appendix A.) The coset leaders were used as the representation. In lattice theoretic terminology, this is called the natural representation.

+	a_2	0	1
b_2	c_2		
0		0	1
1		1	0

G_2.
(a)

+	a_3	0	1	2
b_3	c_3			
0		0	1	2
1		1	2	0
2		2	0	1

G_3.
(b)

Table 3.1.2. A Decomposition of G_6.

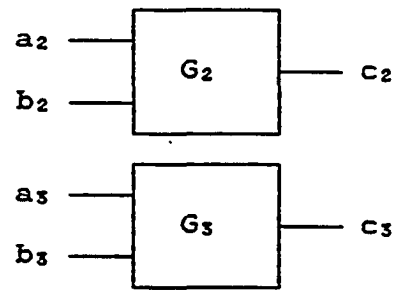


Figure 3.2.
A Parallel Decomposition of G_6.

To analyze the advantage gained by the decomposition, we consider the amount of storage required if the problem were to be realized using a binary memory array and binary coding of the set elements. The original problem G_8 requires a three-bit encoding and 64 locations of memory. Actually, only 36 memory locations are needed, but the binary nature of the encoding requires that the storage size be a power of two. Symmetry in the array is not included in the measure due to the requirements of the comparison and exchange operations. The total number of storage bits to realize G_8 directly is 192. The parallel decomposition, however, requires a one-bit encoding for G_2 with four memory locations and a two-bit encoding for G_3 with 16 memory locations, totaling 36 bits of storage. This results in about a 82% savings in storage.

In Example 3.2 we demonstrate the decomposition method for a serial structure.

Example 3.2. The serial decomposition of G_8.

1) We first generate the subgroups of G_8 as represented in Tables 3.2.1a - 3.2.1d. The resulting S. P. partitions are

$\{\{0, 1, 2, 3, 4, 5, 6, 7\}; \{+, 0, -\}\} \Rightarrow$

$\overline{\{0, 1, 2, 3, 4, 5, 6, 7\}} = \pi_1;$

$\{\{0, 2, 4, 6\}; \{+, 0, -\}\} \Rightarrow \overline{\{0, 2, 4, 6\}; \{1, 3, 5, 7\}} = \pi_1;$

$\{\{0, 4\}; \{+, 0, -\}\} \Rightarrow \overline{\{0, 4\}}; \overline{\{1, 5\}}; \overline{\{2, 6\}}; \overline{\{3, 7\}} = \pi_2;$ and

$\{\{0\}; \{+, 0, -\}\} \Rightarrow \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}; \overline{6}; \overline{7}\} = \pi_0.$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

G_8 = {{0, 1, 2, 3, 4, 5, 6, 7}; {+, 0, -}}.
(a)

+	0	2	4	6
0	0	2	4	6
2	2	4	6	0
4	4	6	0	2
6	6	0	2	4

H_4 = {{0, 2, 4, 6}; {+, 0, -}}.
(b)

+	0	4
0	0	4
4	4	0

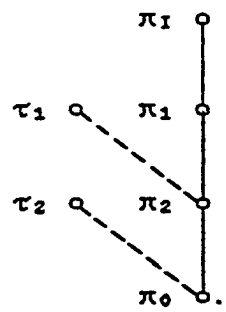
H_2 = {{0, 4}; {+, 0, -}}.
(c)

+	0
0	0

H_1 = {{0}; {+, 0, -}}.
(d)

Table 3.2.1. The Subgroups of G_8.

2) Generate the structure lattice:



3) Select a set of partitions for which the product is π_0.

$\pi_1 \cdot \tau_1 \cdot \tau_2 = \pi_0$, where

$\tau_1 = \{\overline{0, 1, 4, 5}; \overline{2, 3, 6, 7}\}$ and

$\tau_2 = \{\overline{0, 1, 2, 3}; \overline{4, 5, 6, 7}\}$.

Note that the chosen τ's are two-block partitions. It is advantageous to choose partitions with the fewest number of blocks which satisfy the product criteria. By doing so, the representation is minimal.

4) Select a set of representation homomorphisms.

Φ_1: $\{\overline{0, 2, 4, 6}; \overline{1, 3, 5, 7}\} \rightarrow \{\{0, 1\}; \{+, 0, -\}\} = G_2$.

Φ_2: $\{\overline{0, 1, 4, 5}; \overline{2, 3, 6, 7}\} \rightarrow \{\{0, 1\}; \{*\}\} = A_2$;

where * is an operation such that

*: $\{0, 1\}^4 \rightarrow \{0, 1\}$.

Φ_3: $\{\overline{0, 1, 2, 3}; \overline{4, 5, 6, 7}\} \rightarrow \{\{0, 1\}; \{\#\}\} = B_2$;

where # is an operation such that

#: $\{0, 1\}^6 \rightarrow \{0, 1\}$.

The homomorphisms are chosen by assigning elements to each block of each partition. Since π_1 is a S. P. partition, Φ_1 maps the partition to a group of two elements. The

remaining partitions are not S. P. partitions. Therefore, Φ_2 and Φ_3 map to algebras which are serially dependent.

5) The resulting decomposition can be expressed in tabular or block diagram form as expressed in Tables 3.2.2a - 3.2.2c and Figure 3.3, respectively.

The representation homomorphism for G_2 is modulo two addition. A_2 and B_2 are not modulo two operations, but do have "exclusive-or" operations imbedded. The natural representation was used in this example and results in a form called the weighted radix representation. That is, elements belonging to G_8 are represented in binary form and can be restored to their original form by multiplying each element in the new representation by a power of two. Notice that the serialism is created by the iteration of the input variables. This is due to the partitions which do not have the substitution property. The diagram appears very different from the realization achieved using full adders. This results because the carry operation does not possess the substitution property under the group operation. By manipulating the switching expressions for the realization achieved using the lattice method and the realization achieved using full adders, it can be shown that both realizations are Boolean identical and that the carry structure of the full adder is essentially spread through the realization obtained from the lattice method, much like a carry look-ahead circuit.

+	a_0	0	1
b_0	c_0		
0		0	1
1		1	0

$G_2 = \{\{0, 1\}; \{+, 0, -\}\}$.
(a)

*	a_1a_0	00	01	10	11
b_1b_0	c_1				
00		0	0	1	1
01		0	1	1	0
10		1	1	0	0
11		1	0	0	1

$A_2 = \{\{0, 1\}; \{*\}\}$.
(b)

#	$a_2a_1a_0$	000	001	010	011	100	101	110	111
$b_2b_1b_0$	c_2								
000		0	0	0	0	1	1	1	1
001		0	0	0	1	1	1	1	0
010		0	0	1	1	1	1	0	0
011		0	1	1	1	1	0	0	0
100		1	1	1	1	0	0	0	0
101		1	1	1	0	0	0	0	1
110		1	1	0	0	0	0	1	1
111		1	0	0	0	0	1	1	1

$B_2 = \{\{0, 1\}; \{\#\}\}$.
(c)

Table 3.2.2. A Decomposition of G_8.

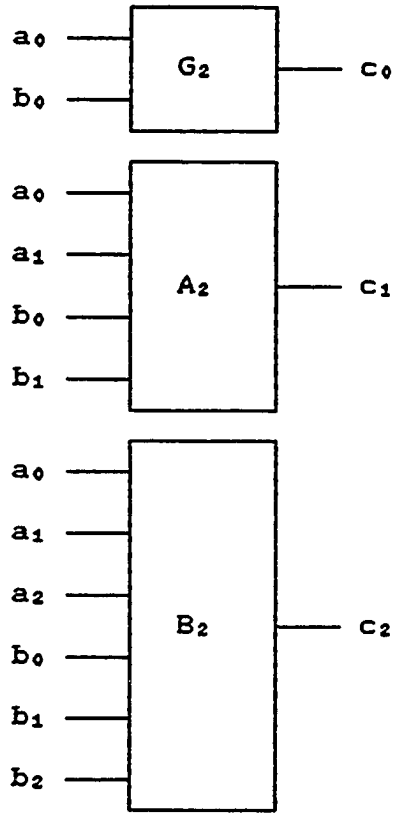


Figure 3.3.
A Serial Decomposition of G_8.

In terms of the measure established in the preceding material, G_8 requires a three-bit encoding and 64 memory locations, for a total of 192 bits of storage. G_2 requires a one-bit encoding and four memory locations; A_2 requires a one-bit encoding and 16 memory locations and B_2 requires a one-bit encoding and 64 memory locations. The total storage requirement for the serial decomposition is 84, resulting in about a 57% savings. A full adder realization of G_8 in terms of storage requirements results in about an 87% savings and thus is more efficient in terms of storage. However, since there is no carry propagation between stages of the lattice theoretic decomposition, the realization can obtain the sum faster.

The two preceding examples were chosen to demonstrate the method for obtaining decompositions for G_n. The method, of course, works for the entire class of G_n and will be generalized in the next section on Z_n. The selection of partitions in the examples was performed in a manner such as to obtain decompositions with desirable characteristics. We will discuss the reasoning behind the selections in the succeeding material.

The Structure of the Ring of Integers Modulo N.

A ring, like a group, is an example of a universal algebra. The operation set F of $R = \{S, F\}$ consists of four mappings:

$f\alpha: \; S^2 \rightarrow S \ni f\alpha(a, b) = a + b, \forall a, b \in S;$

$f\beta$: $S \rightarrow S \ni f\beta(a) = a, \forall a \in S;$

$f\delta$: $S \rightarrow S \ni f\delta(a) = a^{-1}, \forall a \in S$ and

$f\delta$: $S^2 \rightarrow S \ni f\delta(a, b) = a \cdot b, \forall a, b \in S.$

Note that for the ring of integers modulo N there is a fifth mapping which corresponds to the multiplicative identity:

fu: $S \rightarrow S \ni fu(a) = a, \forall a \in S.$

The subalgebra of a ring which satisfies the properties of a universal algebra is an ideal. An ideal $I = \{T; \{+, 0, -, \cdot\}\}$ is defined by the following two properties:

1) Given a ring $R = \{S; \{+, 0, -, \cdot\}\}$ and $I = \{T; \{+, 0, -, \cdot\}\}$, then $H = \{T; \{+, 0, -\}\}$ is a subgroup of $G = \{S; \{+, 0, -\}\}$.

2) Given any $a \in S$ and any $b \in T$ then $a \cdot b \in T$ or $b \cdot a \in T$.

Since an ideal is a subalgebra of a ring then the ideals of a ring form a lattice. To simplify the development of the lattice of ideals of Z_n it can be shown that the lattice of ideals of Z_n is isomorphic to the lattice of subgroups of G_n.

Theorem 3.2. For any ideal

$$I_n = \{T_n; \{+, 0, -, \cdot\}\} \subseteq Z_n = \{S_n; \{+, 0, -, \cdot\}\}$$

there exists a subgroup

$$H_n = \{T_n; \{+, 0, -\}\} \subseteq G_n = \{S_n; \{+, 0, -\}\}.$$

Conversely, for any

$$H_n = \{T_n; \{+, 0, -\}\} \subseteq G_n = \{S_n; \{+, 0, -\}\}$$

there exists an ideal

$$I_N = \{T_N; \{+, 0, -, \cdot\}\} \subseteq Z_N = \{S_N; \{+, 0, -, \cdot\}\}.$$

Proof. The first part of the theorem is proved by observing the definition of an ideal. It is now only necessary to show the converse. A subgroup of $G_N = \{S_N; \{+, 0, -\}\}$, by definition, satisfies the first property of an ideal. Given $H_N = \{T_N; \{+, 0, -\}\} \subseteq G_N$, any element $r \in T_N$ and any element $s \in S_N$, then $|r \cdot s|_N$ is defined to be the addition of r with itself s times, modulo N. By definition, T_N is closed under addition. Therefore, $|r \cdot s|_N \in T_N$ and thus satisfies the second property of an ideal. ∎

As a consequence of Theorems 3.1 and 3.2, it follows that the lattice of ideals of Z_N is isomorphic to the structure lattice of S. P. partitions on Z_N. The same method used for the decomposition of G_N applies to Z_N. In fact, since T_N is an additive subgroup of G_N, steps 1 through 3 of the method generate the same results as those obtained for the decomposition of G_N with the exception that generation of the multiplicative parts of I_N are also required in step 1. In step 5, the decomposition of the multiplicative monoid of Z_N is realized as well as the decomposition of G_N. We demonstrate the method in Examples 3.3 and 3.4 for Z_6 and Z_8 respectively.

Example 3.3. The parallel decomposition of Z_6.

1) As in the method for group decompositions, we first generate the subalgebras. These are ideals as represented

in Tables 3.3.1a - 3.3.1h. From the subalgebras we obtain the following S. P. partitions:

$\{\{0, 1, 2, 3, 4, 5\}; \{+, 0, -, \bullet, 1\}\} \Rightarrow$

$\overline{\{0, 1, 2, 3, 4, 5\}} = \pi_1$;

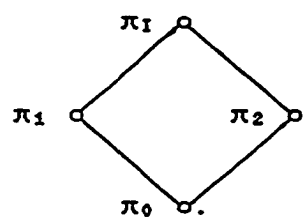
$\{\{0, 2, 4\}; \{+, 0, -, \bullet, 1\}\} \Rightarrow \overline{\{0, 2, 4\}}; \overline{\{1, 3, 5\}} = \pi_1$;

$\{\{0, 3\}; \{+, 0, -, \bullet, 1\}\} \Rightarrow \overline{\{0, 3\}}; \overline{\{1, 4\}}; \overline{\{2, 5\}} = \pi_2$; and

$\{\{0\}; \{+, 0, -, \bullet, 1\}\} \Rightarrow \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}\} = \pi_0$.

Note that the S. P. partitions on Z_8 are the same as those on G_8.

2) Generate the structure lattice.



3) Select a set of partitions for which the product is π_0.

$\pi_1 \bullet \pi_2 = \pi_0$.

4) Select a set of representation homomorphisms:

Φ_1: $\overline{\{0, 2, 4\}}; \overline{\{1, 3, 5\}} \rightarrow \{\{0, 1\}; \{+, 0, -, \bullet, 1\}\} = M_2$.

Φ_2: $\overline{\{0, 3\}}; \overline{\{1, 4\}}; \overline{\{2, 5\}} \rightarrow \{\{0, 1, 2\}; \{+, 0, -, \bullet, 1\}\} = M_3$.

5) We can express the resulting decomposition in tabular form as in Tables 3.3.2a - 3.3.2d or in block diagram form as illustrated by Figure 3.4.

Again, the decomposition results in the Sino-Correspondence. Since the tables and representations are of the same size and type, the savings in storage are

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

G_6 = {{0, 1, 2, 3, 4, 5}; {+, 0, -}}.
(a)

•	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

M_6 = {{0, 1, 2, 3, 4, 5}; {•, 1}}.
(b)

Table 3.3.1. The Ideals of Z_6.

+	0	2	4
0	0	2	4
2	2	4	0
4	4	0	2

$H_3 = \{\{0, 2, 4\}; \{+, 0, -\}\}$.
(c)

•	0	2	4
0	0	0	0
2	0	4	2
4	0	2	4

$K_3 = \{\{0, 2, 4\}; \{\bullet, 1\}\}$.
(d)

+	0	3
0	0	3
3	3	0

$H_2 = \{\{0, 3\}; \{+, 0, -\}\}$.
(e)

•	0	3
0	0	0
3	0	3

$K_2 = \{\{0, 3\}; \{\bullet, 1\}\}$.
(f)

+	0
0	0

$H_1 = \{\{0\}; \{+, 0, -\}\}$.
(g)

•	0
0	0

$K_1 = \{\{0\}; \{\bullet, 1\}\}$.
(h)

Table 3.3.1. The Ideals of Z_6. (continued)

+	a_2	0	1
b_2	c_2		
0		0	1
1		1	0

$G_2 = \{\{0, 1\}; \{+, 0, -\}\}$.
(a)

•	a_2	0	1
b_2	d_2		
0		0	0
1		0	1

$M_2 = \{\{0, 1\}; \{\bullet, 1\}\}$.
(b)

+	a_3	0	1	2
b_3	c_3			
0		0	1	2
1		1	2	0
2		2	0	1

$G_3 = \{\{0, 1, 2\}; \{+, 0, -\}\}$.
(c)

.	a_3	0	1	2
b_3	d_3			
0		0	0	0
1		0	1	2
2		0	2	1

$M_3 = \{\{0, 1, 2\}; \{\bullet, 1\}\}$.
(d)

Table 3.3.2. A Decomposition of Z_6.

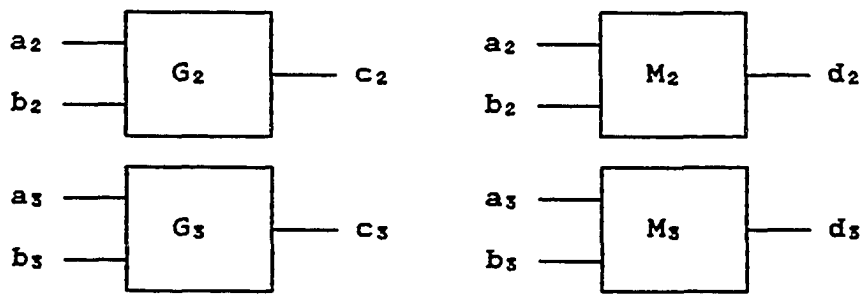


Figure 3.4.
A Parallel Decomposition of Z_6.

the same and theoretically, multiplication can be performed in the same amount of time as addition.

Example 3.4. The serial decomposition of Z_8.

1) Generate the ideals as represented in Tables 3.4.1a - 3.4.1h. The resulting S. P. partitions on Z_8 are

$\{\{0, 1, 2, 3, 4, 5, 6, 7\}; \{+, 0, -, \bullet, 1\} \Rightarrow$

$\overline{\{0, 1, 2, 3, 4, 5, 6, 7\}} = \pi_I;$

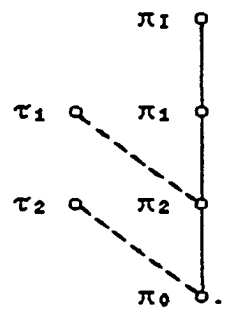
$\{\{0, 2, 4, 6\}; \{+, 0, -, \bullet\}\} \Rightarrow$

$\overline{\{0, 2, 4, 6\}}; \overline{\{1, 3, 5, 7\}} = \pi_1;$

$\{\{0, 4\}; \{+, 0, -, \bullet\}\} \Rightarrow \{\overline{0, 4}; \overline{1, 5}; \overline{2, 6}; \overline{3, 7}\} = \pi_2;$ and

$\{\{0\}; \{+, 0, -, \bullet, 1\}\} \Rightarrow \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}; \overline{6}; \overline{7}\} = \pi_0.$

2) Generate the structure lattice.



3) Select a set of partitions for which the product is π_0.

$\pi_1 \cdot \tau_1 \cdot \tau_2 = \pi_0$ where

$\tau_1 = \{\overline{0, 1, 4, 5}; \overline{2, 3, 6, 7}\}$ and

$\tau_2 = \{\overline{0, 1, 2, 3}; \overline{4, 5, 6, 7}\}.$

4) Select a set of representation homomorphisms.

Φ_1: $\{\overline{0, 2, 4, 6}; \overline{1, 3, 5, 7}\} \rightarrow$

$\{\{0, 1\}; \{+, 0, -, \bullet, 1\}\} = R_2.$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

G_8 = {{0, 1, 2, 3, 4, 5, 6, 7}; {+, 0, -}}.

(a)

•	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	0	2	4	6
3	0	3	6	1	4	7	2	5
4	0	4	0	4	0	4	0	4
5	0	5	2	7	4	1	6	3
6	0	6	4	2	0	6	4	2
7	0	7	6	5	4	3	2	1

M_8 = {{0, 1, 2, 3, 4, 5, 6, 7}; {•, 1}}.

(b)

Table 3.4.1. The Ideals of Z_8.

+	0	2	4	6
0	0	2	4	6
2	2	4	6	0
4	4	6	0	2
6	6	0	2	4

$H_4 = \{\{0, 2, 4, 6\}; \{+, 0, -\}\}$.
(c)

•	0	2	4	6
0	0	0	0	0
2	0	4	0	4
4	0	0	0	0
6	0	4	0	4

$K_4 = \{\{0, 2, 4, 6\}; \{\bullet\}\}$.
(d)

+	0	4
0	0	4
4	4	0

$H_2 = \{\{0, 4\}; \{+, 0, -\}\}$.
(e)

•	0	4
0	0	0
4	0	0

$K_2 = \{\{0, 4\}; \{\bullet\}\}$.
(f)

+	0
0	0

$H_1 = \{\{0\}; \{+, 0, -\}\}$.
(g)

•	0
0	0

$K_1 = \{\{0\}; \{\bullet, 1\}\}$.
(h)

Table 3.4.1. The Ideals of Z_8. (continued)

Φ_2: $\{\overline{0, 1, 4, 5}; \overline{2, 3, 6, 7}\} \rightarrow$

$\{\{0, 1\}; \{*, \Gamma\}\} = C_2$;

where * and Γ are operations such that

*: $\{0, 1\}^4 \rightarrow \{0, 1\}$ and

Γ: $\{0, 1\}^4 \rightarrow \{0, 1\}$.

Φ_3: $\{\overline{0, 1, 2, 3}; \overline{4, 5, 6, 7}\} \rightarrow$

$\{\{0, 1\}; \{\#, \Omega\}\} = D_2$;

where # and Ω are operations such that

#: $\{0, 1\}^6 \rightarrow \{0, 1\}$ and

Ω: $\{0, 1\}^6 \rightarrow \{0, 1\}$.

5) The resulting decomposition can be expressed in tabular form as represented in Tables 3.4.2a - 3.4.2f or in block diagram form as illustrated in Figure 3.5.

The measures of storage efficiency are the same as those obtained for G_8 since the representations are the same and the tables are the same size. Again, the natural representation is the same weighted radix representation as obtained for G_8. In standard algebraic terms this is indicated by the definition of an ideal; i. e., an ideal is a subgroup. An interesting characteristic which results from the use of lattices to obtain the decompositions of Z_8 is the equivalence of the structures of addition and multiplication. The one and only difference is the realization of the components of the decomposition.

+	a_0	0	1
b_0	c_0		
0		0	1
1		1	0

$G_2 = \{\{0, 1\}; \{+, 0, -\}\}$.
(a)

•	a_0	0	1
b_0	d_0		
0		0	0
1		0	1

$M_2 = \{\{0, 1\}; \{\bullet, 1\}\}$.
(b)

*	a_1a_0	00	01	10	11
b_1b_0	c_1				
00		0	0	1	1
01		0	1	1	0
10		1	1	0	0
11		1	0	0	1

$A_2 = \{\{0, 1\}; \{*\}\}$.
(c)

Γ	a_1a_0	00	01	10	11
b_1b_0	d_1				
00		0	0	0	0
01		0	0	1	1
10		0	1	0	1
11		0	1	1	0

$C_2 = \{\{0, 1\}; \{\Gamma\}\}$.
(d)

Table 3.4.2. A Decomposition of Z_8.

#	$a_2a_1a_0$	000	001	010	011	100	101	110	111
$b_2b_1b_0$	c_2								
000		0	0	0	0	1	1	1	1
001		0	0	0	1	1	1	1	0
010		0	0	1	1	1	1	0	0
011		0	1	1	1	1	0	0	0
100		1	1	1	1	0	0	0	0
101		1	1	1	0	0	0	0	1
110		1	1	0	0	0	0	1	1
111		1	0	0	0	0	1	1	1

$B_2 = \{\{0, 1\}; \{\#\}\}$.
(e)

Ω	$a_2a_1a_0$	000	001	010	011	100	101	110	111
$b_2b_1b_0$	d_2								
000		0	0	0	0	0	0	0	0
001		0	0	0	0	1	1	1	1
010		0	0	1	1	0	0	1	1
011		0	0	1	0	1	1	0	1
100		0	1	0	1	0	1	0	1
101		0	1	0	1	1	0	1	0
110		0	1	1	0	0	1	1	0
111		0	1	1	1	1	0	0	0

$D_2 = \{\{0, 1\}; \{\Omega\}\}$.
(f)

Table 3.4.2. A Decomposition of Z_8. (continued)

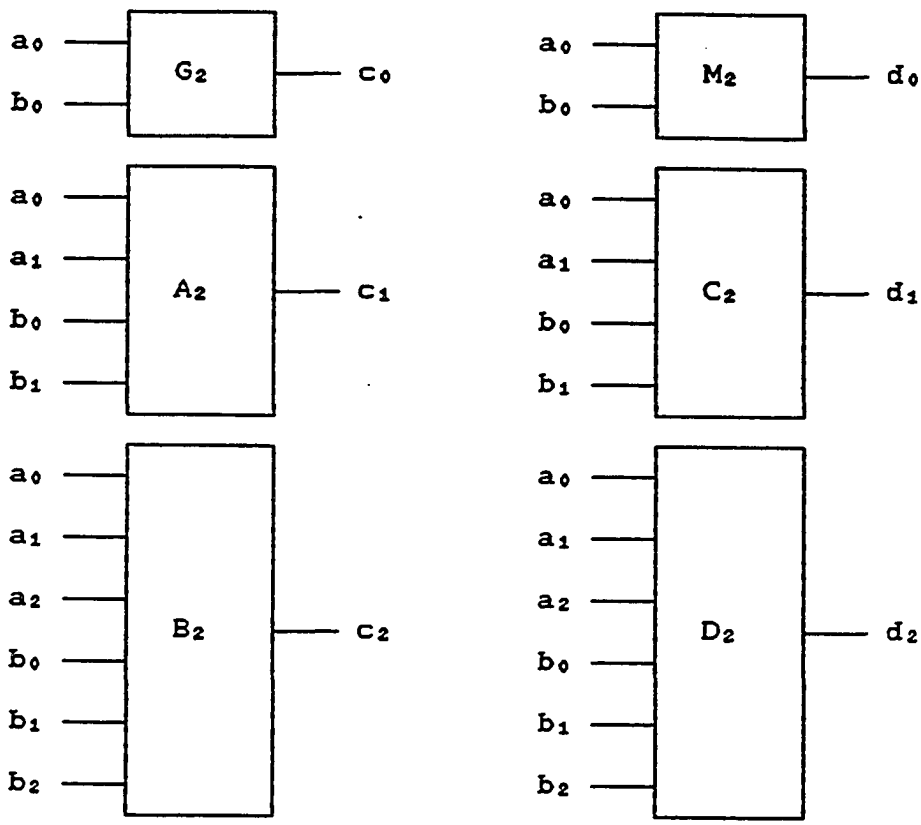


Figure 3.5.
A Serial Decomposition of Z_8.

The General Structure of Z_N.

The ideals of Z_N are directly related to the modulus N. The next theorem provides the means for easily finding the ideals of Z_N and also the subgroups of G_N.

Theorem 3.3. All ideals belonging to Z_N are generated by the divisors of N.

Proof. Assume $N = r \cdot s$. The modulus N in Z_N is congruent to zero under the ring operations. Any divisors of N are called zero-divisors. Therefore, $r \cdot s \equiv 0$.

Generate the set:

$$T_N = \{t \ni t = |r \cdot i|_N, \text{ for } i = 0, \ldots, N - 1\}.$$

Since $r \cdot s \equiv 0$, $r \cdot (s + u) = r \cdot s + r \cdot u = r \cdot u$, for $u = 0, \ldots, s - 1$.

$$T_N = \{t \ni t = |r \cdot u|_N, \text{ for } u = 0, \ldots, s - 1\}.$$

Since $s \leq N$, T_N is of order s.

Form the product

$$|t \cdot u|_N, \ni t \in T_N \text{ and } u \in \{0, \ldots, N - 1\}.$$

Since $t = |r \cdot i|_N$, for $i = 0, \ldots, N - 1$,

$$|t \cdot u|_N = ||r \cdot i|_N \cdot u|_N.$$

But

$$|t \cdot u|_N = ||r \cdot i|_N \cdot u|_N = |r \cdot |i \cdot u|_N|_N \in T_N.$$

Therefore, the external closure property of an ideal is satisfied.

Form the sum $|t_i + t_j|_N$,

$$\ni t_i = |r \cdot i|_N \in T_N, \; t_j = |r \cdot j|_N \in T_N, \text{ and}$$

$$i, j \in \{0, \ldots, s - 1\}.$$

Then

$$|t_i + t_j|_N = ||r \cdot i|_N + |r \cdot j|_N|_N.$$

But

$$||r \cdot i|_N + |r \cdot j|_N|_N = |r \cdot |i + j|_N|_N \in T_N.$$

Hence, T_N is closed under integer addition modulo N.

Since

$$|r \cdot (s - 1)|_N = |r \cdot s - r|_N = |-r|_N,$$

$$|r \cdot (s - 2)|_N = |r \cdot s - 2r|_N = |-2r|_N, \text{ etc.},$$

every element belonging to T_N has an additive inverse. Trivially, O belongs to T_N and is the additive identity. T_N is also associative under the addition operation. Therefore, T_N is a group under addition modulo N and is a subgroup of G_N. Hence, $I_N = \{T_N; \{+, O, -, \cdot\}\}$ is an ideal. ∎

Corollary 3.3. The lattice of the ideals of Z_N is the dual of the lattice of the divisors of N. ∎

The element r is called a generator of the ideal and r = 1 and r = N generate the trivial ideals Z_N and Z_0, respectively.

By Theorems 3.1, 3.2, 3.3 and Corollary 3.3, the lattice of S. P. partitions on Z_N is the dual of the lattice of divisors of N. Each divisor of N corresponds to a generator of an ideal. Thus the structure lattice for Z_N is the lattice of divisors of N inverted. By the Fundamental Theorem of Arithmetic, $N = P_0^{n(0)}P_1^{n(1)}, \ldots, P_N^{n(N)}$, where P_i is a prime, n(i) is some nonnegative integer, and

$P_j \neq P_k$. In lattice terms, each $P_i^{n(i)}$ directly corresponds to a chain of length n(i). That is, beginning with one, each succeeding term is divided by its predecessor, ending with $P_i^{n(i)}$. A typical chain is illustrated in Example 3.4. The multiplication of two powers of primes results in the product of their respective chains in the lattice. The divisor lattice and thus the structure lattice is easily constructed. We now demonstrate the construction of a structure lattice for N = 360.

Example 3.5. The Structure Lattice for Z_{360}.

1) We first express 360 in its product of powers of primes form:

$$360 = 8 \cdot 9 \cdot 5 = 2^3 \cdot 3^2 \cdot 5.$$

2) We then construct chains corresponding to each prime as shown in Figure 3.6.

3) Form the direct product of the chains to construct the divisor lattice for 360. This is illustrated in Figure 3.7.

4) The structure lattice for Z_{360} is the dual of the lattice in Figure 3.7.

The element 1 generates the ideal

$$I_{360} = \{\{0, \ldots, 359\}; \{+, 0, -, \cdot, 1\}\};$$

the element 30 generates the ideal

$$I_{12} = \{\{0, 30, 60, \ldots, 330\}; \{+, 0, -, \cdot, 1\}\};$$

etc.

The structure lattice for Z_n is distributive. The distributive property is used when the lattice is

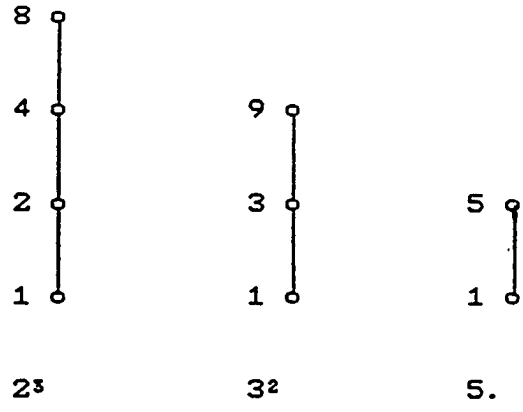


Figure 3.6.
The Prime Divisor Chains for N = 360.

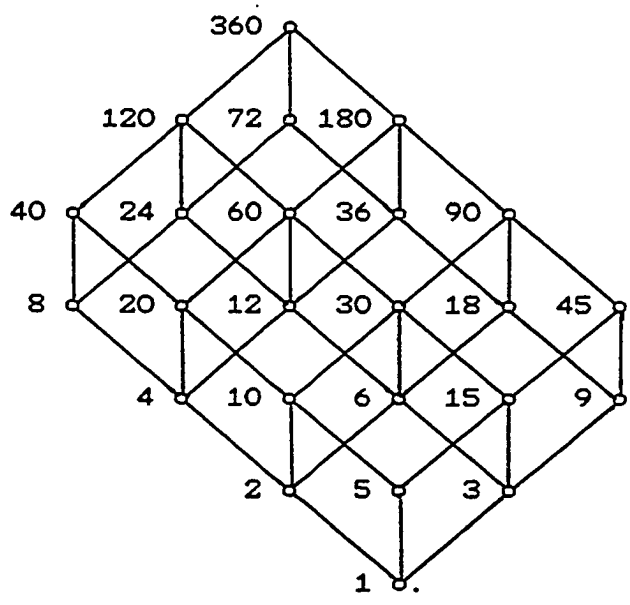


Figure 3.7.
The Divisor Lattice for N = 360.

constructed by taking the product of chains. An important implication of this property is that a decomposition based on the generating chains is irreducible. That is, it can not be directly decomposed further. This suggests the following procedure for obtaining an efficient decomposition of Z_n using lattice theoretic techniques.

A General Method for the Decomposition of Z_n.

1) Express Z_n in its product of powers of primes form.

2) Obtain a parallel decomposition of Z_n, where each parallel block corresponds to Z_{M_i}, where $M_i = P_i^{n(i)}$. If the natural representation is used, the Sino-Correspondence results.

3) Decompose each parallel block into n(i) serial blocks. If the natural representation is used, the weighted radix representation is the result.

Example 3.6. A Lattice Theoretic decomposition of Z_{360}.

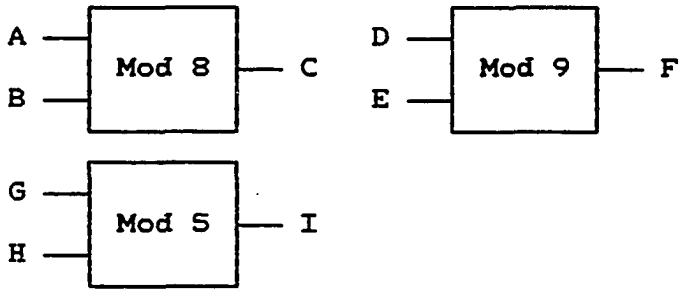
1) Express 360 in its powers of primes form:

$360 = 8 \cdot 9 \cdot 5 = 2^3 3^2 5$.

2) Using the chains and lattice from Example 3.5, the parallel structures shown in Figure 3.8 are obtained.

3) Each parallel block is serially decomposed as illustrated in Figure 3.9.

Using the previously defined measure, Z_{360} would require 2,359,296 bits of storage. The decomposed representation requires 820 bits of storage, amounting to a savings of about 99.96%.

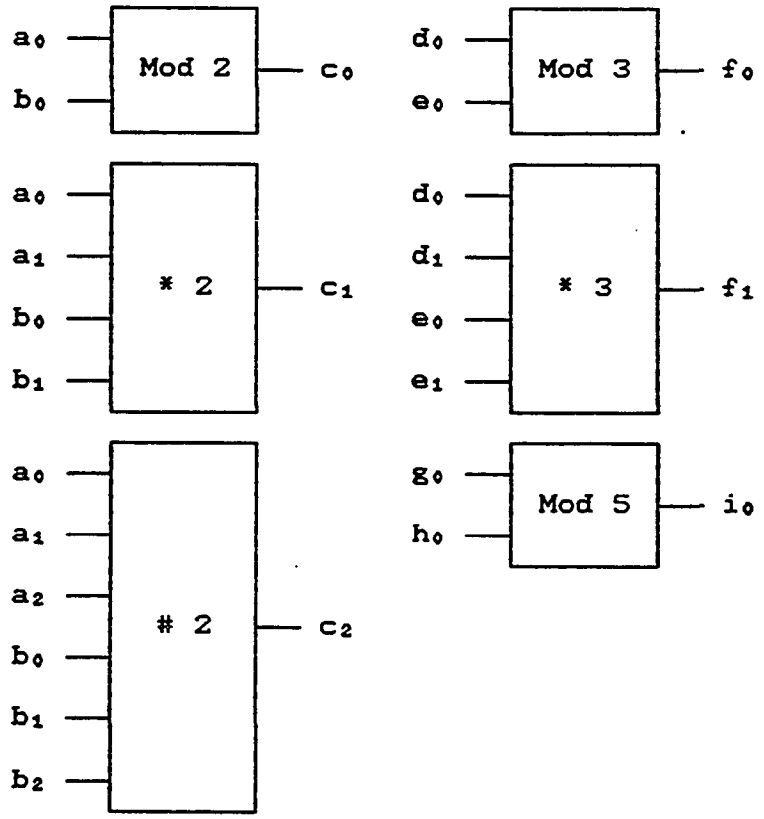


A, B, C $\in$ {0, 1, 2, 3, 4, 5, 6, 7};

D, E, F $\in$ {0, 1, 2, 3, 4, 5, 6, 7, 8};

G, H, I $\in$ {0, 1, 2, 3, 4}.

Figure 3.8.
A Parallel Decomposition of Z_{360}.



$a_2, a_1, a_0, b_2, b_1, b_0, c_2, c_1, c_0 \in \{0, 1\}$;

$d_1, d_0, e_1, e_0, f_1, f_0 \in \{0, 1, 2\}$;

$g_0, h_0, i_0 \in \{0, 1, 2, 3, 4\}$;

* 2: $\{0, 1\}^4 \rightarrow \{0, 1\}$;

2: $\{0, 1\}^6 \rightarrow \{0, 1\}$ and

* 3: $\{0, 1, 2\}^4 \rightarrow \{0, 1, 2\}$.

Figure 3.9.
A Complex Decomposition of Z_{360}.

The amount of storage saved by decomposition into parallel and serial forms tends to increase depending on the composition of N. It must be noted that if N is prime, then Z_N is a field and is not a universal algebra. Therefore, it can not be decomposed using lattice theoretic techniques. It should also be noted that the results of decomposing integer addition and multiplication correspond to similar results derived by Winograd in references [7][8]. However, the use of lattice theoretic method results in a derivation which is simpler and systematic.

In practice, the resulting decompositions do require a systematic representation homomorphism. The homomorphisms chosen for the examples in this chapter are the natural representation homomorphisms. The resulting parallel decompositions are the Sino-Correspondence and do suffer from the lack of overflow detection and a sign flag. Translation to and from the Sino-Correspondence is also not a simple task. However, methods for translation and sign detection are being actively researched. (See reference [17], for example.) The weighted radix representation can be translated using the Euclidean division algorithm. Other representations can be used, but are not considered as part of this investigation. Some of the results presented in this chapter may seem to restate already known results; however, they are established here using lattice theoretic

methods which lay the foundation for their use in the decomposition of problems of higher complexity.

Chapter IV

Polynomial Decomposition

Polynomial methods are used extensively in processing systems. Polynomial expansions have long been used to approximate elementary functions such as sine and exponential functions. To simplify the analysis and synthesis of linear systems, transform techniques are used to reduce the complexity of a problem from manipulation of integro-differential equations to an equivalent algebraic system of polynomials. One of the measures of the complexity of an algorithm is the amount of computational time it takes to execute the algorithm to completion. For a certain class of algorithms, the computational time is on the order of a polynomial. This class is often referred to as a polynomial class or P-class problem. If the computational time is on the order of an exponential, the class is nonpolynomial or NP-complete [24]. This is another example of the use of polynomials in processing systems.

A polynomial is generally defined as an expression:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0,$$

where x is called the indeterminate and the a_i's are called coefficients. The coefficients belong to a structure called an integral domain which, for the sake of simplicity, will

be restricted to a field referred to as the ground field. The reason for this terminology will become apparent later. The degree of the polynomial is denoted by $|P(x)| = n$; i. e., the highest power of an indeterminate with a nonzero coefficient. An alternate representation of a polynomial consists of expressing the polynomial as a vector of coefficients without the indeterminates. For example, given a polynomial $P(x) = x^3 + 2x^2 + 3$, its vector representation is 1 2 0 3.

Polynomial addition is defined as follows:

Given two polynomials,

$$Pa(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0 \text{ and}$$

$$Pb(x) = b_m x^m + b_{m-1} x^{m-1} + \ldots + b_1 x + b_0,$$

their sum is

$$Pa(x) + Pb(x) = \sum_{i=0}^{m} (a_i + b_i) x^i + \sum_{j=m+1}^{n} a_i x^i, \qquad (4.1)$$

where $n > m$.

Similarly, multiplication of polynomials is defined as

$$Pa(x) \cdot Pb(x) = \sum_{i=0}^{n} \sum_{j=0}^{m} a_i b_j x^{i+j}. \qquad (4.2)$$

A set of polynomials, which is closed under polynomial addition and multiplication is a commutative ring. As with integers, a finite or residue class ring can be constructed by performing the ring operations modulo some polynomial, $Pm(x)$. The properties of the ring of polynomials modulo

Pm(x) are analogous to those of the ring of integers modulo N.

The Structure of the Ring of Polynomials Modulo Pm(x)

Let the ring of polynomials modulo Pm(x) be denoted by $R_{Pm(x)} = \{S_{Pm(x)}; \{+, O, -, \cdot\}\}$. $R_{Pm(x)}$ is a universal algebra and the ideals of $R_{Pm(x)}$ form a lattice. To examine the structure of the polynomial ring, it is necessary to show that S. P. partitions can be formed from the ideals of the ring and that the ideals can be generated from the divisors of Pm(x). We demonstrate these two principles in Theorems 4.1 and 4.2.

Theorem 4.1. The lattice of ideals of $R_{Pm(x)}$ is isomorphic to the lattice of S. P. partitions on $R_{Pm(x)}$.

Proof. Since the ideals of $R_{Pm(x)}$ are additive subgroups of $R_{Pm(x)}$, additive cosets can be formed. A subgroup and its cosets have the substitution property by Lemma 3.1. Therefore, S. P. partitions can be formed by adjoining the members of the ideals with their additive cosets. Ideals are partially ordered by set inclusion, as are their additive cosets. This is also the definition of the partial ordering on partitions. Hence, the lattices are isomorphic. ∎

Using Theorem 4.1, the decomposition method developed in Chapters II and III can be used for the decomposition of $R_{Pm(x)}$.

A General Method for the Decomposition of $R_{P_g(x)}$.

1) Find all S. P. partitions or alternately find all ideals and then generate the S. P. partitions from their cosets.

2) Generate the structure lattice.

3) Using the parallel and serial decomposition properties, determine which partitions should be used to derive a decomposition which meets some predetermined criteria.

4) Choose a set of representation homomorphisms.

5) Represent the decomposed polynomial ring using the representation homomorphisms.

We demonstrate the procedure in Example 4.1.

Example 4.1. Decomposition of R_{x^3+1} over GF(2).

1) We generate the ideals of the ring of polynomials modulo $x^3 + 1$ over the field GF(2). They are illustrated in Tables 4.1.1a - 4.1.1h. From the ideals, we construct the S. P. partitions on R_{x^3+1}. Note that the vector representation is used.

{{000, 001, 010, 011, 100, 101, 110, 111}; {+, 0, -, •}} ⇒

$\overline{\{000, 001, 010, 011, 100, 101, 110, 111\}} = \pi_1$.

{{000, 011, 101, 110}; {+, 0, -, •}} ⇒

$\overline{\{000, 011, 101, 110\}}$; $\overline{\{001, 010, 100, 111\}} = \pi_4$.

{{000, 111}; {+, 0, -, •}} ⇒

$\overline{\{000, 111\}}$, $\overline{\{001, 110\}}$, $\overline{\{010, 101\}}$; $\overline{\{100, 011\}} = \pi_2$.

+	000	001	010	011	100	101	110	111
000	000	001	010	011	100	101	110	111
001	001	000	011	010	101	100	111	110
010	010	011	000	001	110	111	100	101
011	011	010	001	000	111	110	101	100
100	100	101	110	111	000	001	010	011
101	101	100	111	110	001	000	011	010
110	110	111	100	101	010	011	000	001
111	111	110	101	100	011	010	001	000

G_{1001} =
{{000, 001, 010, 011, 100, 101, 110, 111}; {+, 0, -}}.
(a)

•	000	001	010	011	100	101	110	111
000	000	000	000	000	000	000	000	000
001	000	001	010	011	100	101	110	111
010	000	010	100	110	001	011	101	111
011	000	011	110	101	101	110	011	000
100	000	100	001	101	010	110	011	111
101	000	101	011	110	110	011	101	000
110	000	110	101	011	011	101	110	000
111	000	111	111	000	111	000	000	111

M_{1001} = {{000, 001, 010, 011, 100, 101, 110, 111}; {•}}.
(b)

Table 4.1.1. The Ideals of R_{1001}.

+	000	011	101	110
000	000	011	101	110
011	011	000	110	101
101	101	110	000	011
110	110	101	011	000

$H_{011} = \{\{000, 011, 101, 110\}; \{+, 0, -\}\}$.
(c)

•	000	011	101	110
000	000	000	000	000
011	000	101	110	011
101	000	110	011	101
110	000	011	101	110

$L_{011} = \{\{000, 011, 101, 110\}; \{\cdot\}\}$.
(d)

+	000	111
000	000	111
111	111	000

$H_{111} = \{\{000, 111\}; \{+, 0, -\}\}$.
(e)

•	000	111
000	000	000
111	000	111

$L_{111} = \{\{000, 111\}; \{\cdot\}\}$.
(f)

+	000
000	000

$H_{000} = \{\{000\}; \{+, 0, -\}\}$.
(g)

•	000
000	000

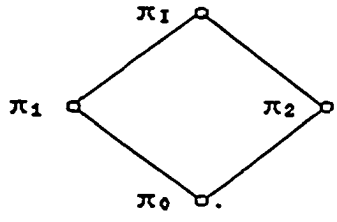
$L_{000} = \{\{000\}; \{\cdot\}\}$.
(h)

Table 4.1.1. The Ideals of R_{1001}. (continued)

$\{\{000\}; \{+, 0, -, \cdot\}\} \Rightarrow$

$\{\overline{000}; \overline{001}; \overline{010}; \overline{011}; \overline{100}; \overline{101}; \overline{110}; \overline{111}\} = \pi_0$.

2) Generate the structure lattice:



3) Select a set of partitions such that their product is π_0.

$\pi_1 \cdot \pi_2 = \pi_0$.

4) Select a set of representation homomorphisms.

Φ_1: $\{\overline{000, 011, 101, 110}; \overline{001, 010, 100, 111}\} \rightarrow$

$\{\{0, 1\}; \{+, 0, -, \cdot, 1, ^{-1}\}\} = GF(2)$.

Φ_2: $\{\overline{000, 111}; \overline{001, 110}; \overline{010, 101}; \overline{100, 011}\} \rightarrow$

$\{\{00, 01, 10, 11\}; \{+, 0, -, \cdot, 1, ^{-1}\}\} = GF(2^2)$.

5) The decomposition can be represented in tabular form as shown in Tables 4.1.2a - 4.1.2d or in block diagram form as illustrated in Figure 4.1.

Several observations can be made about Example 4.1. Using the measure established in Chapter III, the original table requires 192 bits of storage. The tables obtained via the parallel decomposition require 36 bits of storage, amounting to a savings of approximately 81%.

Note that the resulting decomposition produces a Galois field of two elements GF(2), which is the ground field, and an extension field $GF(2^2)$. The total number of ground field operations can be used to establish another measure. To

+	a_0	0	1
b_0	c_0		
0		0	1
1		1	0

Addition group of GF(2).
(a)

•	d_0	0	1
e_0	f_0		
0		0	0
1		0	1

Multiplication group of GF(2).
(b)

+	a_1	00	01	10	11
b_1	c_1				
00		00	01	10	11
01		01	00	11	10
10		10	11	00	01
11		11	10	01	00

Addition Group of $GF(2^2)$.
(c)

•	d_1	00	01	10	11
e_1	f_1				
00		00	00	00	00
01		00	01	10	11
10		00	10	11	01
11		00	11	01	10

Multiplication group of $GF(2^2)$.
(d)

Table 4.1.2. A Decomposition of R_{1001}.

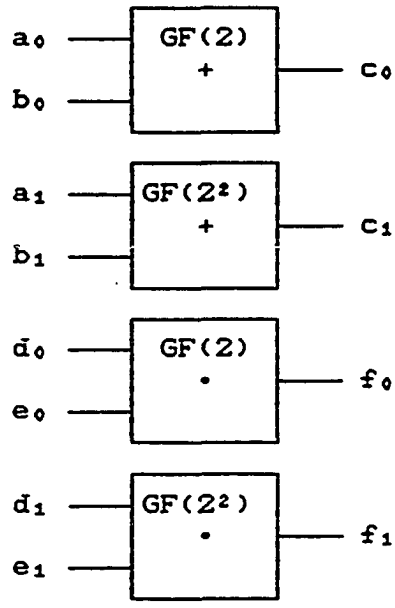


Figure 4.1.
A Parallel Decomposition of $R_x{}^3{}_{+1}$.

multiply two polynomials in R_{x^3+1}, nine ground field multiplications and six ground field additions are required. The representation derived from the decomposition requires one ground field multiplication and one extension field multiplication. The extension field multiplication can be further broken down to four ground field multiplications and three ground field additions. The total number of ground field multiplications and additions are reduced by the chosen representation.

The General Structure of $R_{P_m(x)}$

To generalize the theory of polynomial ring decomposition, it is necessary to show the relationship between the modulus and the ideals of a polynomial ring. We demonstrate this relationship in Theorem 4.2.

Theorem 4.2. All ideals belonging to $R_{P_m(x)}$ are generated by the divisors of Pm(x).

Proof. The proof of Theorem 4.2 is analogous to that of Theorem 3.3. The factors of Pm(x) are zero divisors.

Let

$$Pm(x) = r(x) \cdot s(x).$$

Then

$$|Pm(x)| = |r(x)| + |s(x)|.$$

Generate the set

$$T_{P_m(x)} = \{t(x) \ni t(x) = |r(x) \cdot i(x)|_{P_m(x)}\}$$

for $i(x) = 0, \ldots, Pm\text{-}1(x)$.

Pm-1(x) is the largest polynomial of degree less than |Pm(x)|.

Since r(x) • s(x) ≡ O,

$$|r(x) \bullet [s(x) + u(x)]|_{P_m(x)} =$$

$$|r(x) \bullet s(x) + r(x) \bullet u(x)|_{P_m(x)}$$

for u(x) = O, ..., q(x);

where q(x) is the largest polynomial of degree less than |s(x)|.

$$T_{P_m(x)} = \{t(x) \ni t(x) = |r(x) \bullet i(x)|_{P_m(x)}\}$$

for i(x) = O, ..., q(x).

This implies that the order of $T_{P_m(x)}$ is n • |s(x)|, where n is the order of the ground field.

Form the product

$$|t(x) \bullet u(x)|_{P_m(x)}$$

such that

$$t(x) \in T_{P_m(x)} \text{ and } u(x) \in \{O, \ldots, Pm\text{-}1(x)\}.$$

Since $t(x) = |r(x) \bullet i(x)|_{P_m(x)}$,

$$|t(x) \bullet u(x)|_{P_m(x)} = ||r(x) \bullet i(x)|_{P_m(x)} \bullet u(x)|_{P_m(x)},$$

for i(x) = O, ..., Pm-1(x).

By associativity of the modulo operation,

$$|t(x) \bullet u(x)|_{P_m(x)} =$$

$$|r(x) \bullet |i(x) \bullet u(x)|_{P_m(x)}|_{P_m(x)} \in T_{P_m(x)}.$$

Therefore, the external closure property of an ideal is satisfied.

Form the sum $|t_i(x) + t_j(x)|_{P_m(x)}$, such that

$$t_i(x) = |r(x) \bullet i(x)|_{P_m(x)} \in T_{P_m(x)}, \text{ and}$$

$t_j(x) = |r(x) \cdot j(x)|_{Pm(x)} \in T_{Pm(x)}$, for

$i(x), j(x) \in \{0, \ldots, q(x)\}$.

Then

$$|t_i(x) + t_j(x)|_{Pm(x)} =$$

$$||r(x) + i(x)|_{Pm(x)} + |r(x) + j(x)|_{Pm(x)}|_{Pm(x)}.$$

But

$$||r(x) + i(x)|_{Pm(x)} + |r(x) + j(x)|_{Pm(x)}|_{Pm(x)} =$$

$$|r(x) \cdot |i(x) + j(x)|_{Pm(x)}|_{Pm(x)} \in T_{Pm(x)}.$$

Thus, $T_{Pm(x)}$ is closed under polynomial addition modulo Pm(x).

Since

$$|r(x) \cdot [s(x) - 1]|_{Pm(x)} =$$

$$|r(x) \cdot s(x) - r(x)|_{Pm(x)} = |-r(x)|_{Pm(x)},$$

$$|r(x) \cdot [s(x) - x]|_{Pm(x)} =$$

$$|r(x) \cdot s(x) - x \cdot r(x)|_{Pm(x)} = |-x \cdot r(x)|_{Pm(x)}, \text{ etc.},$$

every element belonging to $T_{Pm(x)}$ has an additive inverse. Trivially, zero is a member of $T_{Pm(x)}$ and is the additive identity. $T_{Pm(x)}$ is also associative under addition and thus is an ideal generated by r(x). ∎

From Theorem 4.2, a general structure theory for polynomial rings can be established which is the analog to the structure theory developed for integer rings. Given any modulus Pm(x), it can be expressed as a product of powers of irreducible polynomials, each of which generates an ideal in the ring of polynomials modulo Pm(x). Each linear factor (polynomial factors of degree one) corresponds to a parallel

block in the decomposition which is isomorphic to the ground field. Nonlinear, irreducible factors correspond to parallel blocks which are isomorphic to extension fields generated by the irreducible polynomials. Any repeated factors result in a serial decomposition. Hence, a general method for the decomposition of a polynomial ring is suggested.

A General Method for the Decomposition of $R_{p_s(x)}$.

1) For a given modulus, factor the modulus into powers of irreducible polynomials.

2) Each factor corresponds to a parallel component which can be realized as a structure isomorphic to the ground field or an extension field.

3) Repeated factors are then serially decomposed.

Applications of Polynomial Decompositions

Consider a modulus of the form $x^n - 1$. In the ring of polynomials modulo $x^n - 1$, $x^n \equiv 1$. Therefore, multiplying any two polynomials belonging to the ring modulo $x^n - 1$ is essentially the cyclic convolution of two sequences represented by the coefficients of the multiplier and multiplicand polynomials. Using the lattice theoretic results derived above, the sequences can be represented in a different form as determined by the divisors of the modulus. The following example provides insight into the connection between cyclic convolution and the lattice theoretic representation.

Example 4.2. Decomposition of R_x^{2+2} over GF(3).

1) Generate the ideals belonging to R_x^{2+2}. Only the multiplicative monoids, Tables 4.2.1a - 4.2.1d, will be generated since convolution involves polynomial multiplication. The corresponding S. P. partitions are

{{00, 01, 02, 10, 11, 12, 20, 21, 22}; {•}} ⇒

$\overline{\{00, 01, 02, 10, 11, 12, 20, 21, 22\}} = \pi_I$.

{{00, 11, 22}; {•}} ⇒

$\overline{\{00, 11, 22\}}; \overline{\{01, 12, 20\}}; \overline{\{02, 10, 21\}} = \pi_1$.

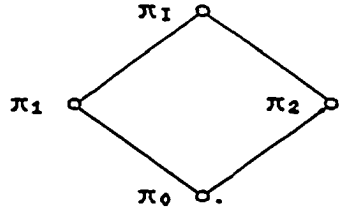
{{00, 12, 21}; {•}} ⇒

$\overline{\{00, 12, 21\}}; \overline{\{01, 10, 22\}}; \overline{\{02, 11, 20\}} = \pi_2$.

{{00}; {•}} ⇒

$\overline{00}; \overline{01}; \overline{02}; \overline{10}; \overline{11}; \overline{12}; \overline{20}; \overline{21}; \overline{22} = \pi_0$.

2) Generate the structure lattice:



3) Choose a set of partitions such that their product is π_0.

$\pi_1 \cdot \pi_2 = \pi_0$.

4) Select a set of representation homomorphisms.

Φ_1: $\overline{\{00, 11, 22\}}; \overline{\{01, 12, 20\}}; \overline{\{02, 10, 21\}} \rightarrow$

{{0, 1, 2}; {•, 1, $^{-1}$}} = multiplication group of GF(3).

•	00	01	02	10	11	12	20	21	22
00	00	00	00	00	00	00	00	00	00
01	00	01	02	10	11	12	20	21	22
02	00	02	01	20	22	21	10	12	11
10	00	10	20	01	11	21	02	12	22
11	00	11	22	11	22	00	22	00	11
12	00	12	21	21	00	12	12	21	00
20	00	20	10	02	22	12	01	21	11
21	00	21	12	12	00	21	21	12	00
22	00	22	11	22	11	00	11	00	22

M_{102} = {{00, 01, 02, 10, 11, 12, 20, 21, 22}; {•}}.
(a)

•	00	11	22
00	00	00	00
11	00	22	11
22	00	11	22

L_{11} = {{00, 11, 22}; {•}}.
(b)

•	00	12	21
00	00	00	00
12	00	12	21
21	00	21	12

L_{12} = {{00, 12, 21}; {•}}.
(c)

•	00
00	00

L00 = {{00}; {•}}.
(d)

Table 4.2.1. The Submonoids of M_{102}.

Φ_2: $\{\overline{00, 12, 21}; \overline{01, 10, 22}; \overline{02, 11, 20}\} \rightarrow$

$\{\{0, 1, 2\}; \{\cdot, 1, {}^{-1}\}\}$ = multiplication group of GF(3).

5) The decomposition can be represented in tabular form as shown in Tables 4.2.2a - 4.2.2b.

Multiply two polynomials, e. g.,

Conventional Representation	Lattice Theoretic Representation
$x + 1$	2 0
$2x + 2$	1 0
$2x + 2$	2 0.
$2x + 2$	
$x + 1$	

Note that the lattice theoretic representation resulted in a term by term multiplication. This resembles performing a cyclic convolution by taking the Fourier transform or in the finite case, the number theoretic transform and multiplying term by term. Using lattice theoretic methods, we can construct a transform matrix which is similar to a number theoretic transform matrix.

In Example 4.2 the set of homomorphisms are chosen such that their images are the natural representation. Since the divisors of the modulus are linear and are not repeated, a parallel decomposition results and the representation is the Sino-Correspondence for polynomials. This implies that for each polynomial in the ring, a representation can be obtained by finding the residue of the polynomial modulo each factor of the modulus of the ring. However, if each

•	a_0	0	1	2
b_0	c_0			
0		0	0	0
1		0	1	2
2		0	2	1

Multiplication group of GF(3).
(a)

•	a_1	0	1	2
b_1	c_1			
0		0	0	0
1		0	1	2
2		0	2	1

Multiplication group of GF(3).
(b)

Table 4.2.2. A Decomposition of M_{102}.

factor is linear and is a zero divisor, then each factor $(x - a_i)$ is equivalent to zero. Thus, the representation of a polynomial can be obtained by evaluating the polynomial for each root of the modulus. This homomorphic transformation process can be represented in vector-matrix form as shown in equation 4.1.

$$\begin{bmatrix} c_0 \\ \cdot \\ \cdot \\ \cdot \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} a_0^0 & \cdots & a_0^{n-1} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{n-1}^0 & \cdots & a_{n-1}^{n-1} \end{bmatrix} \begin{bmatrix} b_0 \\ \cdot \\ \cdot \\ \cdot \\ b_{n-1} \end{bmatrix}. \qquad (4.1)$$

In equation 4.1, the a_i's are roots of the modulus, the b_j's are the coefficients of the polynomial to be represented, and the c_j's are the coefficients of the representation. We recall that the coefficients belong to a ground field. If successive powers of an element belonging to the ground field generate all non-zero members of the field then the element is called primitive. If the a_i's in equation 4.1 can be generated by a primitive element α, then equation 4.1 can be written as a summation:

$$c(i) = \sum_{j=0}^{n-1} b(j) \cdot \alpha^{ij} . \qquad (4.2)$$

Equation 4.2 is a Fourier transform over the ground field.

Given R_{x^n-1}, if $x^n - 1$ can be factored into linear factors over the ground field, then each factor corresponds

to a root of unity. Each factor generates an ideal from which the residue class ring is constructed. The residue class ring corresponds directly to the discrete Fourier transform. Even if the factors are nonlinear and irreducible, a transform can still be obtained by using the extension fields created by the nonlinear factors. A familiar example is the Fourier transform of a sequence of real numbers producing a sequence of complex numbers.

When the ground field is finite, the transform is designated number theoretic. Number theoretic transforms typically require that the term α in equation 4.2 be primitive. This produces a maximum length transform matrix in terms of powers of α. The lattice theoretic form produces a transform matrix which consists of rows of powers of each root and is derived without having to adhere to the restriction that a primitive element exists as with the typical number theoretic transform. Thus, it is more systematic in its approach.

Note that if the ground field is the field of complex numbers, the lattice theoretic result produces the conventional discrete Fourier transform. Thus lattice theory unifies the concept of performing a circular convolution using the Fourier transform independent of the ground field. The process used to obtain the above result is important because it demonstrates that the discrete Fourier transform can be derived without having to resort to

the idea of sampling the continuous Fourier transform. That is, the derivation is performed using discrete algebraic theory only.

The above results agree with those demonstrated by R. Blahut [9], but the lattice theoretic approach provides a simpler and systematic method. Dr. Blahut argues that there are many similarities between the techniques used in digital signal processing and error control coding and that the differences were a result of a "historical accident." He uses the discrete Fourier transform as the unifying influence. The use of the discrete Fourier transform in signal processing is well known. In error control coding cyclic codes have been analyzed and synthesized using polynomials referred to as Mattson-Solomon polynomials [19]. Dr. Blahut demonstrated that Mattson-Solomon polynomials result from taking the discrete Fourier transform of the code vectors in a finite field [25].

Note that cyclic codes can be represented as a generator or G matrix and a parity or H matrix, such that the product of G and H^T produce a matrix containing all zeros [19]. An H matrix for a typical code is of the form illustrated in equation 4.3 where α_i's are roots of the generator polynomial, n is the code word length and r is the degree of the generator polynomial. The H matrix in equation 4.3 is in the same form as the representation obtained from the lattice theoretic method for polynomial

$$H = \begin{bmatrix} \alpha_1^{n-1} & \alpha_1^{n-2} & \cdots & \alpha_1^{1} & \alpha_1^{0} \\ \alpha_2^{n-1} & \alpha_2^{n-2} & \cdots & \alpha_2^{1} & \alpha_2^{0} \\ \vdots & \vdots & & \vdots & \vdots \\ \alpha_r^{n-1} & \alpha_r^{n-2} & \cdots & \alpha_r^{1} & \alpha_r^{0} \end{bmatrix}. \qquad (4.3)$$

rings with moduli of the form $x^n - 1$. We have demonstrated the relation between this representation and the discrete Fourier transform. Hence, the relation between the discrete Fourier transform and certain types of error control codes is demonstrated using lattice theoretic techniques.

Chapter V

Fourier Transforms and Linear Systems

In the preceding sections we have shown that there are fundamental structures for integer and polynomial rings which can be derived using lattice theoretic methods. In this chapter, these same methods are used to demonstrate the decomposition properties of the discrete Fourier transform (DFT) and discrete linear systems. That is, fast Fourier transform (FFT) algorithms can be derived and linear systems can be decomposed into parallel and serial structures in a systematic manner through the use of S. P. partitions and their resulting lattices.

Besides the obvious gain in systematicity, results which augment the unification arguments of this research are derived. One result is the common basis of the Cooley-Tukey FFT (CTFFT) algorithm [5] and the Good-Thomas FFT (GTFFT) algorithm [20][21]. It was at first argued that the two algorithms were equivalent. It was subsequently observed that this was a mistaken assumption [22]. It will be shown here that, in fact, the two algorithms are results of serial and parallel decompositions of an integer modulus. This will become apparent, once lattice theoretic methods are applied to the problem. Lattice theory will also be used to

generalize the partial fraction expansion of a linear system as the fundamental method of decomposition regardless of the field over which the system is realized.

DFT to FFT

The FFT algorithms are well known to signal processing engineers as important and significant results in computational complexity. The following treatise will parallel that found in references on signal processing. (See [18], for example.) The DFT can be expressed as a summation:

$$X(j) = \sum_{k=0}^{N-1} x(k) \cdot W_N^{jk}, \qquad (5.1)$$

where the x(k)'s are the discrete samples in the time domain, the X(j)'s are samples in the frequency domain, N is the number of samples, $W_N = e^{2\pi i/N}$ and j and k are indices such that $0 \leq j, k < N$. The DFT can also be expressed in matrix form:

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix} = \begin{bmatrix} W^{0\cdot 0} & W^{0\cdot 1} & \cdots & W^{0\cdot(N-1)} \\ W^{1\cdot 0} & W^{1\cdot 1} & \cdots & W^{1\cdot(N-1)} \\ \vdots & \vdots & \cdots & \vdots \\ W^{(N-1)\cdot 0} & W^{(N-1)\cdot 1} & \cdots & W^{(N-1)\cdot(N-1)} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}.$$

Since W^{ij} has period N, the multiplication of the indices i and j is performed modulo N. This corresponds directly to the monoid portion of the ring of integers modulo N. From previous discussions in Chapter III we can easily see that N can be factored into powers of primes and that for each prime there is an associated parallel block representation which can be further decomposed serially if the exponent of the prime is greater than one. The derivation of the CTFFT and GTFFT algorithms depend on manipulation of the representation of the indices. The procedure for obtaining FFT algorithms is essentially the same procedure for obtaining decompositions of integer rings with the addition of a step which expresses the indices in terms of the representations resulting from the decomposition. The following two examples will illustrate how both the CTFFT and GTFFT algorithms can be systematically derived using lattice theoretic methods.

Example 5.1. Radix Eight DFT.

1) We first write the product of the indices in tabular form, illustrated by Table 5.1. This is the monoid portion of Z_8. Thus, S. P. partitions can be derived from the ideals of Z_8:

$\{\{0, 1, 2, 3, 4, 5, 6, 7\}; \{\cdot\}\} \Rightarrow$

$\overline{\{0, 1, 2, 3, 4, 5, 6, 7\}} = \pi_1$;

$\{\{0, 2, 4, 6\}; \{\cdot\}\} \Rightarrow \overline{\{0, 2, 4, 6\}}; \overline{\{1, 3, 5, 7\}} = \pi_1$;

$\{\{0, 4\}; \{\cdot\}\} \Rightarrow \overline{\{0, 4\}}; \overline{\{1, 5\}}; \overline{\{2, 6\}}; \overline{\{3, 7\}} = \pi_2$; and

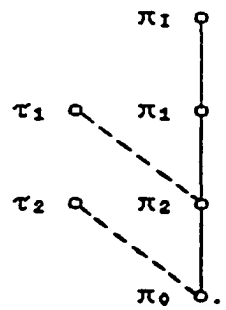
•	k	0	1	2	3	4	5	6	7
j									
0		0	0	0	0	0	0	0	0
1		0	1	2	3	4	5	6	7
2		0	2	4	6	0	2	4	6
3		0	3	6	1	4	7	2	5
4		0	4	0	4	0	4	0	4
5		0	5	2	7	4	1	6	3
6		0	6	4	2	0	6	4	2
7		0	7	6	5	4	3	2	1

$|j \cdot k|_8$.

Table 5.1. The Monoid M_8.

$\{\{0\}, \{\cdot\}\} \Rightarrow \{\overline{0};\ \overline{1};\ \overline{2};\ \overline{3};\ \overline{4};\ \overline{5};\ \overline{6};\ \overline{7}\} = \pi_0$.

2) Generate the structure lattice:



3) Find a set of partitions such that their product is π_0.

$$\pi_1 \cdot \tau_1 \cdot \tau_2 = \pi_0,$$

where

$\tau_1 = \{\overline{0, 1, 4, 5};\ \overline{2, 3, 6, 7}\}$ and

$\tau_2 = \{\overline{0, 1, 2, 3};\ \overline{4, 5, 6, 7}\}$.

4) Generate the set of natural representation homomorphisms, in this case a weighted radix representation.

Φ_1: $\{\overline{0, 2, 4, 6};\ \overline{1, 3, 5, 7}\} \rightarrow \{\{0, 1\}; \{\cdot\}\} = M_2$;

Φ_2: $\{\overline{0, 1, 4, 5};\ \overline{2, 3, 6, 7}\} \rightarrow \{\{0, 1\}; \{\Gamma\}\} = C_2$ and

Φ_3: $\{\overline{0, 1, 2, 3};\ \overline{4, 5, 6, 7}\} \rightarrow \{\{0, 1\}; \{\Omega\}\} = D_2$;

where

Γ: $\{0, 1\}^4 \rightarrow \{0, 1\}$ and

Ω: $\{0, 1\}^6 \rightarrow \{0, 1\}$.

5) Express the indices j and k in terms of the new representation:

$$j = 4j_2 + 2j_1 + j_0 \text{ and}$$

$$k = 4k_2 + 2k_1 + k_0. \qquad (5.2)$$

Since $j \cdot k = N$, multiply the weighted radix representations:

$$j \cdot k = 16j_2k_2 + 8j_2k_1 + 8j_1k_2 + 4j_1k_1 + 4j_2k_0 + 4j_0k_2 + 2j_0k_1 + 2j_1k_0 + j_0k_0. \qquad (5.3)$$

All terms which are multiples of eight are equivalent to zero since the operations are performed in Z_8. Equation 5.3 can be rearranged as follows:

$$j \cdot k = k_2(4j_0) + k_1(4j_1 + 2j_0) + k_0(4j_2 + 2j_1 + j_0). \qquad (5.4)$$

Substituting equation 5.4 into equation 5.1 we obtain

$$X(j) = \sum_{k=0}^{7} x(k) \cdot W_8^{k_2(4j_0)+k_1(4j_1+2j_0)+k_0(4j_2+2j_1+j_0)}. \qquad (5.5)$$

Rearranging the order of the k_i's and dividing by $N = 8$:

$$X(j) = \sum_{k_0=0}^{1} W_8^{k_0 j} \sum_{k_1=0}^{1} W_4^{k_1|j|_4} \sum_{k_2=0}^{1} W_2^{k_2|j|_2} x(k_2,k_1,k_0). \qquad (5.6)$$

Equation 5.6 is the CTFFT in decimation in frequency form. By rearranging the summations the decimation in time form can be obtained. It is important to note that the lattice used to decompose the indices is a chain which indicates a serial decomposition. This is reflected by the presence of the rotation or "twiddle" factors, which are combined with the radix two DFT roots of unity in equation 5.6. The first level of computation consists of eight radix two DFT's depending only on k_2 and j_0. The next two levels depend on k_1, j_1, j_0 and k_0, j_2, j_1, j_0, respectively. This

form concurs with the serial nature of the decomposition as it was developed in Chapter III.

Example 5.2. Radix Six DFT.

1) Write the indices of the DFT in tabular form as displayed in Table 5.2. Since this is the monoid of Z_6, S. P. partitions and their associated lattice can be obtained from the ideals of Z_6.

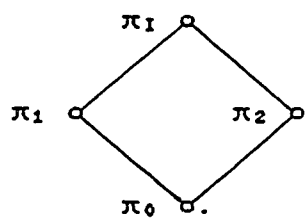
$\{\{0, 1, 2, 3, 4, 5\}; \{\bullet\}\} \Rightarrow \overline{\{0, 1, 2, 3, 4, 5\}} = \pi_I$;

$\{\{0, 2, 4\}; \{\bullet\}\} \Rightarrow \overline{\{0, 2, 4}; \overline{1, 3, 5\}} = \pi_1$;

$\{\{0, 3\}; \{\bullet\}\} \Rightarrow \overline{\{0, 3}; \overline{1, 4}; \overline{2, 5\}} = \pi_2$; and

$\{\{0\}; \{\bullet\}\} \Rightarrow \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}\} = \pi_0$.

2) Generate the structure lattice from the partitions:



3) Select a set of partitions such that their product is π_0.

$\pi_1 \cdot \pi_2 = \pi_0$.

4) Generate the set of natural representation homomorphisms, in this case the Sino-Correspondence.

Φ_1: $\overline{\{0, 2, 4}; \overline{1, 3, 5\}} \rightarrow \{\{0, 1\}; \{\bullet\}\} = M_2$.

Φ_2: $\{\overline{0, 3}; \overline{1, 4}; \overline{2, 5}\} \rightarrow \{\{0, 1, 2\}; \{\bullet\}\} = M_3$.

5) Expressing the indices j and k in terms of the new representation using the Chinese Remainder Theorem produces

•	k	0	1	2	3	4	5
j							
0		0	0	0	0	0	0
1		0	1	2	3	4	5
2		0	2	4	0	2	4
3		0	3	0	3	0	3
4		0	4	2	0	4	2
5		0	5	4	3	2	1

$|j \cdot k|_6$.

Table 5.2. The Monoid M_6.

$j = 3j_2 + 4j_3$ and

$$k = 3k_2 + 4k_3. \qquad (5.7)$$

Multiplying j and k, we obtain

$$j \cdot k = 16j_3k_3 + 12j_2k_3 + 12j_3k_2 + 9j_2k_2. \qquad (5.8)$$

All multiples of six are equivalent to zero since the multiplication is performed in Z_6. Equation 5.8 is now substituted into equation 5.1.

$$X(j) = \sum_{k=0}^{5} x(k) \cdot W_6^{16j_3k_3 + 9j_2k_2}. \qquad (5.9)$$

Splitting the summation and finding the greatest common divisors (GCD's) of N = 6 and the constant multipliers we obtain

$$X(j_2,j_3) = \sum_{k_2=0}^{1} W_2^{3j_2k_2} \sum_{k_3=0}^{2} W_3^{8j_3k_3} x(k_2,k_3). \qquad (5.10)$$

Note that $|3|_2 = 1$ and $|8|_3 = 2$. Therefore,

$$X(j_2,j_3) = \sum_{k_2=0}^{1} W_2^{j_2k_2} \sum_{k_3=0}^{2} W_3^{2j_3k_3} x(k_2,k_3). \qquad (5.11)$$

This example is an FFT algorithm, but is not an example of the GTFFT algorithm since there is a multiplier, other than one, in the exponent of the radix three transform. Note that this constant is not a rotation factor. To obtain an example of the GTFFT algorithm, i. e., remove the constant, a different representation is required for one of the indices. This requires a different representation homomorphism. The homomorphism Φ_1 cannot be changed without destroying the kernal which violates the definition of a

morphism. Φ_2 however can be changed while preserving the kernal. Permuting the representation Φ_2 we obtain

$$\Phi_2': \{\overline{0, 3}; \overline{1, 4}; \overline{2, 5}\} \rightarrow \{\{0, 2, 1\}; \{\cdot\}\} = M_3'.$$

Using Φ_2', we can represent j as follows:

$$j = 3j_2 + 2j_3. \tag{5.12}$$

Multiplying j and k results in

$$j \cdot k = 12j_3k_3 + 9j_3k_2 + 8j_2k_3 + 6j_2k_2. \tag{5.13}$$

Again, multiples of N = 6 are equal to zero, leaving

$$X(j) = \sum_{k=0}^{5} x(k) \cdot W_6^{9j_3k_2 + 8j_2k_3}. \tag{5.14}$$

Dividing the exponents by the GCD's of the exponents and N and breaking the summation produces

$$X(j_2,j_3) = \sum_{k_2=0}^{1} W_2^{3j_3k_2} \sum_{k_3=0}^{2} W_3^{4j_2k_3} x(k_2,k_3). \tag{5.15}$$

But $|3|_2 = 1$ and $|4|_3 = 1$. Therefore,

$$X(j_2,j_3) = \sum_{k_2=0}^{1} W_2^{j_3k_2} \sum_{k_3=0}^{2} W_3^{j_2k_3} x(k_2,k_3). \tag{5.16}$$

The above equation is an example of the GTFFT algorithm. Some interesting characteristics may be observed. In its traditional development, the GTFFT algorithm requires that the factors of N be relatively prime. The lattice theoretic derivation satisfies this criterion without specifically requiring it. There are no rotation factors or constant multipliers. This agrees with the parallel nature of the representation. In terms of the lattice theoretic approach, this is the first example

discussed in this dissertation of a representation homomorphism which is not the natural representation homomorphism. This indicates that there may exist homomorphisms other than the natural ones which possess desirable characteristics. In the above example the new homomorphism was obtained by a simple permutation of the natural representation.

Examples 5.1 and 5.2 suggest a method for finding a general FFT algorithm for a highly composite N. As described in Chapter III, N can be expressed as powers of primes which directly correspond to a lattice consisting of the product of prime chains. For each prime raised to a power there will exist a parallel block; i. e., the DFT of N elements can be decomposed into a GTFFT based on the primes raised to a power as factors of N. Since the factors are powers of primes, they are obviously relatively prime. For each prime raised to a power greater than one, the radix p^i DFT can be serially decomposed into a CTFFT. This follows the method derived in Chapter III for the decomposition of integer rings and demonstrates that lattice theory is the basis for the decomposition of integers and therefore DFT's.

Discrete Linear Systems

Discrete linear systems are used in digital signal processing, automatic control and error control coding. They are thus very important to the systems science aspect

of Electrical Engineering. Decomposition is a useful tool in the analysis and synthesis of such systems. This section will demonstrate that lattice theory provides a systematic means of decomposing linear systems.

For the sake of simplicity, the systems used as examples will be constrained to single input, single output, causal, discrete and linear. Such a system can be represented as a transfer function in the z domain. That is, the system, input sequence and output sequence are represented as polynomials with indeterminate z and coefficients over some predetermined ground field. Thus, the system can be represented:

$$T(z) = \frac{a_m z^m + a_{m-1} z^{m-1} + \ldots + a_0 z^0}{b_n z^n + b_{n-1} z^{n-1} + \ldots + b_0 z^0} = \frac{A(z)}{B(z)}$$

where n is generally greater than m. Therefore, the z transform of the output sequence

$$Y(z) = \frac{A(z)}{B(z)} X(z), \qquad (5.17)$$

where X(z) is the z transform of the input sequence. Shift registers can be used to realize the simultaneous multiplication and division of a polynomial by numerator and denominator polynomials. (See [19].) An important observation is that the registers in such a realization contain the state of the system at any given time. This corresponds to the following equation:

$$S(z) = \left| A(z)\ X(z) \right|_{B(z)}, \qquad (5.18)$$

where S(z) is the state polynomial.

Equation 5.18 implies that the set of state polynomials form a ring modulo the denominator polynomial B(z). Using this information, the method derived in Chapter IV for the decomposition of polynomial rings can be used to decompose the state polynomials and therefore, the systems. From Chapter IV we know that a polynomial can be factored into irreducible polynomials, some of which may be repeated. Each irreducible, nonlinear factor can be factored further via its extension field. The polynomials may then be represented by the Sino-Correspondence for each factor raised to a power. For factors which are repeated, the weighted radix representation for polynomials can be used. These forms correspond respectively to the parallel and serial decompositions as obtained via the structure lattice of the polynomial ring.

Let the state polynomials be represented in Sino-Correspondence form. The original polynomials can be recovered by the Chinese Remainder Theorem for polynomials. Thus,

$$S(z) = \left| A(z)\ X(z) \right|_{B(z)} \quad \text{or}$$

$$S(z) = \left| \sum_{i=0}^{m-1} \frac{B(z)}{Bi(z)} \right| \left| Wi(z)Si(z) \right|_{Bi(z)} \Big|_{B(z)} \qquad (5.19)$$

where m is the number of relatively prime factors,

$$\prod_{i=0}^{m-1} Bi(z) = B(z), \tag{5.20}$$

$$Si(z) = \left| S(z) \right|_{Bi(z)} \quad \text{and} \tag{5.21}$$

$$\left| \frac{B(z)}{Bi(z)} Wi(z) \right|_{Bi(z)} = 1. \tag{5.22}$$

Let the input equal the impulse; i. e., let X(z) = 1. Then the output of the system is the impulse response, which is sufficient information to describe the operation of the system. Then

$$S(z) = \left| A(z) \right|_{B(z)}. \tag{5.23}$$

Assuming the order of A(z) is less than that of B(z), then A(z) = S(z). Therefore,

$$\frac{S(z)}{B(z)} = \frac{A(z)}{B(Z)}. \tag{5.24}$$

Substituting in equation 5.19 and diving by B(z),

$$\frac{A(z)}{B(z)} = \frac{\left| \sum_{i=0}^{m-1} \frac{B(z)}{Bi(z)} \left| Wi(z)Ai(z) \right|_{Bi(z)} \right|_{B(z)}}{B(z)}. \tag{5.25}$$

However, B(z) can be canceled on the right side of equation 5.25, yielding

$$\frac{A(z)}{B(z)} = \left| \sum_{i=0}^{m-1} \frac{1}{Bi(z)} \left| Wi(z)Ai(z) \right|_{Bi(z)} \right|_{B(z)}. \tag{5.26}$$

If left in summation form, the outside modulo operation can

be dropped, since each term will be of order less than B(z). The inside modulo operation can be dropped since the Ai(z) are formed modulo Bi(z) and the Wi(z) are divided by Bi(z). Under these assumptions equation 5.26 becomes

$$\frac{A(z)}{B(z)} = \sum_{i=0}^{m-1} \frac{Wi(z)Ai(z)}{Bi(z)} . \qquad (5.27)$$

Equation 5.27 is easily recognized as a partial fraction expansion of T(z), where the Wi(z)Ai(z)'s are the residues.

A similar argument can be made for the weighted radix representation. Given a repeated root $C(z)^n$ and using the state arguments developed in the above discussion, the state polynomial can be represented in weighted radix form, where the C(z) is the radix. Thus

$$A(z) = \sum_{i=0}^{n-1} C(z)^i \, Ai(z), \qquad (5.28)$$

where the Ai(z)'s are the polynomials derived from the representation homomorphism for a chain. Dividing both sides of equation 5.28 by B(z) we obtain

$$\frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{n-1} C(z)^i \, Ai(z)}{B(z)} . \qquad (5.29)$$

Equation 5.29 is the partial fraction expansion of a repeated root. An interesting observation about equation 5.29 is that it does not contain the differentiation process which is typically used in the expansion of a repeated root. Note also, that no dependence on a particular ground field

exists in the developments of both the parallel and serial expansions. To demonstrate the method and that it works, regardless of the field, we consider Example 5.3.

Example 5.3. Decomposition of a Linear System Over GF(2).

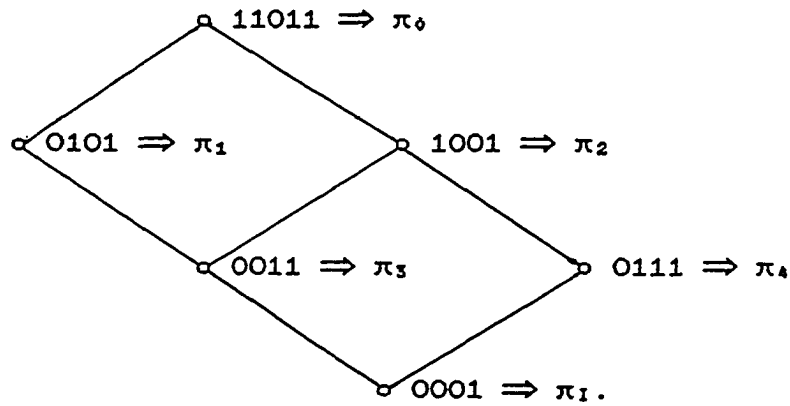
Let

$$\frac{A(z)}{B(z)} = \frac{z^3 + z + 1}{z^4 + z^3 + z + 1} = \frac{z^3 + z + 1}{(z + 1)^2 (z^2 + z + 1)}.$$

In vector representation,

$$\frac{A(z)}{B(z)} = \frac{1011}{11011}.$$

The divisor lattice for the ring modulo B(z) = 11011 is



It can be easily be seen from the lattice that there will be two parallel blocks, one of which can be decomposed serially. The partitions associated with each divisor are as follows:

$$\pi_I = \{\overline{0000,\ 0001,\ 0010,\ 0011,\ 0100,\ 0101,\ 0110,\ 0111};\ \overline{1000,\ 1001,\ 1010,\ 1011,\ 1100,\ 1101,\ 1110,\ 1111}\};$$

π_4 = {$\overline{0000, 0111, 1110, 1001}$; $\overline{0001, 0110, 1111, 1000}$;
$\overline{0010, 0101, 1100, 1011}$; $\overline{0011, 0100, 1101, 1010}$};

π_3 = {$\overline{0000, 0011, 0110, 0101, 1100, 1111, 1010, 1001}$;
$\overline{0001, 0010, 0111, 0100, 1101, 1110, 1011, 1000}$};

π_2 = {$\overline{0000, 1001}$; $\overline{0001, 1000}$; $\overline{0010, 1011}$; $\overline{0011, 1010}$;
$\overline{0100, 1101}$; $\overline{0101, 1100}$; $\overline{0110, 1111}$; $\overline{0111, 1110}$};

π_1 = {$\overline{0000, 0101, 1010, 1111}$; $\overline{0001, 0100, 1011, 1110}$;
$\overline{0010, 0111, 1000, 1101}$; $\overline{0011, 0110, 1001, 1100}$};

and

π_0 = {$\overline{0000}$; $\overline{0001}$; $\overline{0010}$; $\overline{0011}$; $\overline{0100}$; $\overline{0101}$; $\overline{0110}$; $\overline{0111}$;
$\overline{1000}$; $\overline{1001}$; $\overline{1010}$; $\overline{1011}$; $\overline{1100}$; $\overline{1101}$; $\overline{1110}$; $\overline{1111}$}.

Note again that the polynomials are expressed in vector notation. From the lattice it is evident that $\pi_4 \cdot \pi_1 = \pi_0$. However, a two-block partition τ can be constructed such that $\pi_3 \cdot \tau = \pi_4$. Let

τ = {$\overline{0000, 0101, 1010, 1111, 0010, 0111, 1000, 1101}$;
$\overline{0001, 0100, 1011, 1110, 0011, 0110, 1001, 1100}$}.

Perform a parallel decomposition using π_4 and π_1. Let

$$\pi_1 \Rightarrow Q1(z) = \left| \begin{array}{c} A(z) \\ \\ 0101 \end{array} \right| \quad \text{and}$$

$$\pi_4 \Rightarrow Q4(z) = \left| \begin{array}{c} A(z) \\ \\ 0111 \end{array} \right| .$$

That is, the blocks of π_1 are represented by 00, 01, 10, and 11 respectively, as are the blocks of π_4. Since the above representation is in Sino-Correspondence form, the CRT can

be used to to recover the original polynomials. Therefore,

$$A(z) = \left| (0111)\cdot P1(z)\cdot Q1(z) + (0101)\cdot P4(z)\cdot Q4(z) \right|_{B(z)},$$

where

$$\left| (0111)\cdot P1(z) \right|_{0101} = 1 \text{ and}$$

$$\left| (0101)\cdot P4(z) \right|_{0111} = 1.$$

The polynomials P4(z) and P1(z) can easily be found by using the representations derived from the partitions. Let A(z) = 0111, then Q1(z) = 10 and Q4(z) = 00. Substituting and solving for P1(z)

$$0111 = \left| (0111)\cdot P1(z)\cdot (10) \right|_{11011},$$

which implies

$$\left| P1(z)\cdot (10) \right|_{11011} = 1.$$

Therefore, P1(z) = 1101. Similarly, P4(z) = 1101.

Substituting the results,

$$A(z) = \left| (0111)\cdot Q1(z) + (1111)\cdot Q4(z) \right|_{11011}.$$

By using the partitions τ and π_3, Q1(z) can be represented in weighted radix form:

$$Q1(z) = (0011)\cdot Q3(z) + Q\tau(z).$$

Substituting for Q1(z) and dividing by B(z) = 11011,

$$\frac{A(z)}{B(z)} = \frac{(0111)\cdot(0011)\cdot Q3(z)}{11011} + \frac{(0111)\cdot Q\tau(z)}{11011} + \frac{(1111)\cdot Q4(z)}{11011}.$$

Since A(z) = 1011; Q3(z) = 1, Qτ(z) = 1 and Q4(z) = 10.

Factoring B(z) and cancelling like terms produces

$$\frac{A(z)}{B(z)} = \frac{1}{0011} + \frac{1}{0101} + \frac{110}{0111}.$$

In indeterminate form,

$$\frac{A(z)}{B(z)} = \frac{1}{z+1} + \frac{1}{z^2+1} + \frac{z^2+z}{z^2+z+1}.$$

Example 5.3 illustrates two important points about partial fraction expansions. First, there is no dependence on the particular ground field. If there are factors of the denominator which are irreducible over the ground field, the natural representation produces elements which belong to an extension field without having to generate the field and split the polynomial. Second, differentiation was not used in the example or in the development of the method. The method, using a lattice theoretic approach, was strictly algebraic. Again, it has been demonstrated that lattice theory is basic to the decomposition process and provides a single method which can be used in a variety of problems.

Chapter VI

Results, Conclusions, and Future Research

Summary of Results and Future Research

Previous reseachers have shown that lattice theory is an effective tool for the analysis of the structures of a great number of system design problems. This dissertation demonstrates further that the use of lattice theory provides a basis for a diversity of problems involving processing systems. The unifying nature of the lattice theoretic method presented here allows a nontraditional interpretation of the fundamental characteristics of systems.

Specifically, the ring of integers modulo N, Z_N, is studied. Using lattice theory, a general method for decomposing Z_N into parallel and serial structures in the form of the Sino-Correspondence and the weighted radix representation is derived. These structures are obtained by using the natural representation homomorphisms. A topic for further investigation is the significance of structures obtained by using representations other than the natural ones. In terms of the decomposition properties for integers, it is demonstrated that if an integer ring is implemented as a table look-up a substantial savings in memory can be obtained for a highly composite N. It is also

demonstrated that, theoretically, integer addition and multiplication can be performed in the same amount of time if implemented in decomposed table look-up form.

The lattices for Z_n display the distributive property which contributes to an irreducible decomposition if the proper partitions are chosen. Another topic for future research involves the implications of choosing partitions which do not produce irreducible decompositions, i. e., representations which contain redundancies. Is there any connection between redundancies obtained by selecting reducible decompositions and error control coding for arithmetic processes? Arithmetic error control codes are constructed using integer ideals and the Sino-Correspondence [19][26]. The lattice of ideals is an integral part of the decomposition procedure for integers and the Sino-Correspondence is an example of a natural representation of an integer decomposition. It follows that the lattice theoretic method may be a useful tool in the construction of arithmetic error control codes.

The structure of polynomial rings is investigated in Chapter IV. The structure of polynomial rings is shown to be analogous to that of integers with the exception of extension fields. The Sino-Correspondence and the weighted radix representation for polynomials are the results of decomposition using the natural representation homomorphisms. Given an irreducible, nonlinear factor of

the modulus, extension fields are the result of the natural representation.

An important result in polynomial decomposition is obtained for a modulus of the form $x^n - 1$. Multiplication of polynomials modulo $x^n - 1$ is essentially the convolution of two ground field sequences. Using the lattice theoretic approach to the convolution problem, the discrete Fourier transform is derived directly without resorting to the continuous Fourier transform or sampling theory. The discrete Fourier transform is, in fact, a special case of the Sino-Correspondence. Thus the Chinese Remainder Theorem for irreducible divisors of $x^n - 1$ is the inverse Fourier transform. By-products of this result are the independence of the discrete Fourier transform on the ground field over which it is implemented and thus the unnecessary distinction created by the term number theoretic transform. What is the significance to signal processing, if any, of performing polynomial addition and multiplication using a modulus other than $x^n - 1$?

Similar questions arise when the same techniques are applied to error control codes. Parity matrices for error control codes can be derived in a similar manner using lattice theory. An important topic for future research is, again, the use of representations other than the natural ones. That is, given a code matrix developed by lattice theoretic techniques, can different codes be obtained by

selecting representation homomorphisms other than the natural ones?

Chapter V exploits further the results of Chapters III and IV to derive FFT algorithms and decompose discrete linear systems. A general method of decomposing the FFT for composite N is derived using the lattice theoretic approach. The Sino-Correspondence and weighted radix representation once again result as the natural representations for the GTFFT and CTFFT algorithms, respectively. However, the first example of a representation other than the natural one is used in the GTFFT decomposition which again points to further investigations into other representations. It is important to note that the lattice theoretic method of derivation is more systematic than that used for the original derivation of the FFT algorithms.

When applied to discrete linear systems, lattice theory reveals that a partial fraction expansion is the natural decomposition. Each irreducible factor of the denominator of the transfer function describing the system represents a parallel block in the decomposition with repeated factors being further decomposed into serial structures. The representations are the Sino-Correspondence and weighted radix representation. One of the interesting consequences of this process is that the method of derivation does not depend on the concept of a derivative for repeated roots. This fact and the results derived for the discrete Fourier

transform present another area for future research. How can continuous systems and discrete systems be treated in an analogous manner using a lattice theoretic approach?

One of the practical constraints in the use of lattice theoretic techniques is that the representations be systematic. This can be overcome by using the algebra of relations which is a lattice ordered monoid [10]. Instead of realizing systems using conventional Boolean algebra, relational algebra could be used. This is suggested by Birkhoff [23] because of the structure which is prevalent in the algebra of relations. It is a more systematic approach, but suffers from the set theoretic representation of its elements. A set theoretic representation is realized by a binary vector where each bit corresponds to one element. Perhaps with a more advanced technology it will become a feasible approach.

To restrict the scope of this dissertation to a practical level, the structures studied are groups and rings. Monoids and semigroups are examples of universal algebras, but are less structured. Thus, finding the structure lattices becomes more tedious. However, the process can be automated by computer and it is therefore feasible to obtain profitable results for a greater variety of systems. Higher order structures such as matrix algebras are also possible candidates for the application of lattice theoretic techniques.

To conclude, the use of lattice theory provides a universal tool for obtaining fundamental structure information in a variety of disciplines. Many of the subdisciplines within the field of Electrical Engineering have a common basis in the structures derived using lattice theoretic methods. This dissertation demonstrates the versatility of the use of lattice theoretic methods for representative applications in the area of processing systems.

References

[1] H. A. Curtis, The Design of Switching Circuits. Princeton, NJ: D. Van Nostrand Company, Inc., 1962.

[2] Z. Kohavi, Switching and Finite Automata Theory. New York: McGraw-Hill Book Company, 1978.

[3] J. Hartmanis and R. E. Stearns, Algebraic Structure Theory of Sequential Machines. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1966.

[4] J. C. Abbott, Editor, Trends in Lattice Theory. New York: Van Nostrand Reinhold Company, 1970.

[5] J. W. Cooley and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Mathematics of Computation, Vol. 19, No. 90, pp. 297-301, 1965.

[6] S. Winograd, "On Computing the Discrete Fourier Transform," Proceedings of the National Academy of Science, pp. 1005-1006, 1976.

[7] S. Winograd, "On the Time Required to Perform Addition," Journal of the Association for Computing Machinery, Vol. 12, pp. 277-285, 1965.

[8] S. Winograd, "On the Time Required to Perform Multiplication," Journal of the Association for Computing Machinery, Vol. 14, pp. 793-802, 1967.

[9] R. Blahut, "Algebraic Fields, Signal Processing, and Error Control," Proceedings of the IEEE, Vol. 73, No. 5, pp. 874-893, 1985.

[10] G. Birkhoff, Lattice Theory, American Mathematical Society Colloquium Publications, Vol. XXV. Providence, Rhode Island: American Mathematical Society, 1967.

[11] G. Birkhoff and T. C. Bartee, Modern Applied Algebra. New York: McGraw-Hill Book Company, 1970.

[12] P. Crawley and R. P. Dilworth, Algebraic Theory of Lattices. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1973.

[13] G. Szasz, Introduction to Lattice Theory. New York: Academic Press, Budapest: The Publishing House of the Hungarian Academy of Sciences, 1963.

[14] M. Aigner, Combinatorial Theory. New York: Springer-Verlag New York Inc., 1979.

[15] G. Birkhoff and S. Mac Lane, A Survey of Modern Algebra. New York: The Macmillan Company, 1953.

[16] N. Szabo and R. Tanaka, Residue Arithmetic and Its Applications to Computer Technology. New York: McGraw-Hill, 1967.

[17] T. Van Vu, "Efficient Implementations of the Chinese Remainder Theorem for Sign Detection and Residue Decoding," IEEE Trans. on Computers, Vol. C-34, No. 7, pp. 646-651, 1985.

[18] L. Rabiner and B. Gold, Theory and Application of Digital Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, Inc., 1975.

[19] W. Peterson and E. Weldon, Jr., Error Correcting Codes. Cambridge, Massachusetts: The MIT Press, 1972.

[20] I. Good, "The Interaction Algorithm and Practical Fourier Analysis," J. Roy. Statist. Soc., Ser. B, Vol. 20, pp. 361-372, 1958.

[21] L. Thomas, "Using a Computer to Solve Problems in Physics," Application of Digital Computers. Boston, Mass.: Ginn, 1963.

[22] J. Cooley, P. Lewis, and P. Welch, "Historical Notes on the Fast Fourier Transform," IEEE Trans. Audio Electroacoust., Vol. AU-15, pp. 76-79, 1967.

[23] G. Birkhoff and M. Hall, Jr., Editors, Computers in Algebra and Number Theory, Vol. IV, SIAM-AMS Proceedings. Providence, Rhode Island: American Mathematical Society, 1971.

[24] A. Aho, J. Hopcroft, and J. Ullman, The Design and Analysis of Computer Algorithms. Reading, Mass.: Addison-Wesley Publishing Company, 1975.

[25] R. Blahut, "Transform Techniques for Error Control Codes," IBM Journal of Research and Development., Vol. 23, No. 3, pp. 299-315, 1979.

[26] T. R. N. Rao, Error Coding for Arithmetic Processes. New York: Academic Press, 1974.

Appendix

The Sino-Correspondence and the Chinese Remainder Theorem

This appendix provides a brief review of the Sino-correspondence and the Chinese Remainder Theorem. A complete reference is found in Residue Arithmetic and Its Application to Computer Technology by Szabo and Tanaka [16].

Number systems are typically described by a weighted radix representation. That is, a number can be represented by an N-tuple of digits. The position of each digit corresponds to the product of the weight of that digit and a power of the radix which is unique to that position. Hence the term weighted radix. For example, a positive integer k in a radix r representation appears as

$$k = w_{n-1} \ldots w_1 w_0 \Rightarrow$$

$$k = w_{n-1} r^{n-1} + \ldots + w_1 r^1 + w_0 r^0,$$

where $0 \leq w_i < r$. The Euclidean division algorithm can be used to convert a number from one radix to another. The conversion is performed by successively dividing the given number by the new radix. After each division the residue is retained and the quotient becomes the dividend for the next step until the quotient is zero.

The Sino-correspondence or residue representation of a number is also an N-tuple. However the residue

representation differs from the weighted radix because the base consists of N radices: $m_1, \ldots, m_N$. Each radix, m_i, is called a modulus. For a set of moduli, a number can be represented by finding the residue of the number with respect to each of the moduli. The residues are placed in some arbitrary order to form an N-tuple. If the moduli are pair-wise relatively prime, the representation is irredundant and the maximum number representable is the product of the moduli minus one.

Residue representations can be added, subtracted or multiplied by performing the respective operation on corresponding pairs of digits. That is, digits with the same modulus are added, subtracted or multiplied. Any overflows from the operation on the digit are ignored. There are no carries from one modulus to another making this a good representation for fast arithmetic operations. For example, given the moduli 2 and 3 then

$|4|_2 = 0$ and $|4|_3 = 1$.

$|5|_2 = 1$ and $|5|_3 = 2$.

$|4|_6 + |5|_6 = |3|_6 \Rightarrow (0, 1) + (1, 2) = (1, 0)$.

The number which has the residue representation of (1, 0) for moduli of 2 and 3 respectively is 3.

To convert a number in residue representation back to a weighted radix representation it is necessary to use the Chinese Remainder Theorem.

The Chinese Remainder Theorem

Given a set of relatively prime moduli $m_1 \cdot \ldots \cdot m_n = M$, then

$$\left| x \right|_M = \left| \sum_{j=1}^{N} m'_j \left| r_j \cdot m'^{-1}_j \right|_{m_j} \right|_M,$$

where $m'_j = M / m_j$. As an example let $M = 6$, $m_1 = 2$ and $m_2 = 3$. Then for $r_1 = 1$ and $r_2 = 0$,

$m'_1 = 3$ and

$m'_2 = 2$.

Therefore,

$m'^{-1}_1 = 1$ and

$m'^{-1}_2 = 1$.

$|r_1 \cdot m'^{-1}_1|_2 = |1 \cdot 1|_2 = 1$

$|r_2 \cdot m'^{-1}_2|_3 = |0 \cdot 2|_3 = 0$.

Thus,

$$| x |_6 = | 3 \cdot 1 + 2 \cdot 0 |_6 = 3.$$

As stated in the above material the Sino-correspondence or residue representation is very suitable for fast arithmetic. Converting between a weighted radix representation and the residue representation is systematic, but not simple. Thus in a practical implementation, design trade-offs must be considered.

Autobiographical Statement

David Livingston was born on September 18, 1954 in Norfolk, Va. He received the BSE. degree in 1976 and ME. degree in 1978 from Old Dominion University. The following is a list of his publications:

D. L. Livingston and M. R. Varanasi, "A Microprocessor Implementation of a Data Compression Algorithm", Virginia Academy of Science, May 1977.

F. T. Kozuh, D. L. Livingston and T. C. Spillman, "System/370 Capability in a Desktop Computer", IBM Systems Journal, Vol. 23, No. 3, 1984.

D. L. Livingston, D. J. Sucher and B. M. Walk, "Multiplexed Addresses for a Two-Card Processor/Memory System, IBM Technical Disclosure Bulletin, Vol. 26, No. 10A, March 1984.

———, "Transparent Hardware Address Offset for Use in a Common Memory, Multiprocessor Environment, IBM Technical Disclosure Bulletin, Vol. 26, No. 11, April 1984.

———, "Synchronization of Peripheral Devices Via Bus Cycle Delay", IBM Technical Disclosure Bulletin, Vol. 28, No. 2, July 1985.

———, Bit Substitution for Refresh of Systems with Unequal Memory and Bus Widths", IBM Technical Disclosure Bulletin, Vol. 27, No. 3, August 1984.

———, "Apparatus and Method for Effecting Dynamic Address Translation in a Microprocessor Implemented Data Processing System", IBM Patent Application, October 1983.

Mr. Livingston has held the National Aeronautics and Space Administration Aeronautical Fellowship, Doctoral Teaching Fellowship and various other research assistantships and is currently employed as a Staff Engineer at IBM Endicott. He is a member of Eta Kappa Nu.